Spediz. in abbonamento postale GR. II/70 L. 2.000

18 CORSO PRATICO COL COMPUTER

è una iniziativa
FABBRI EDITORI
in collaborazione con
BANCO DI ROMA
e OLIVETTI

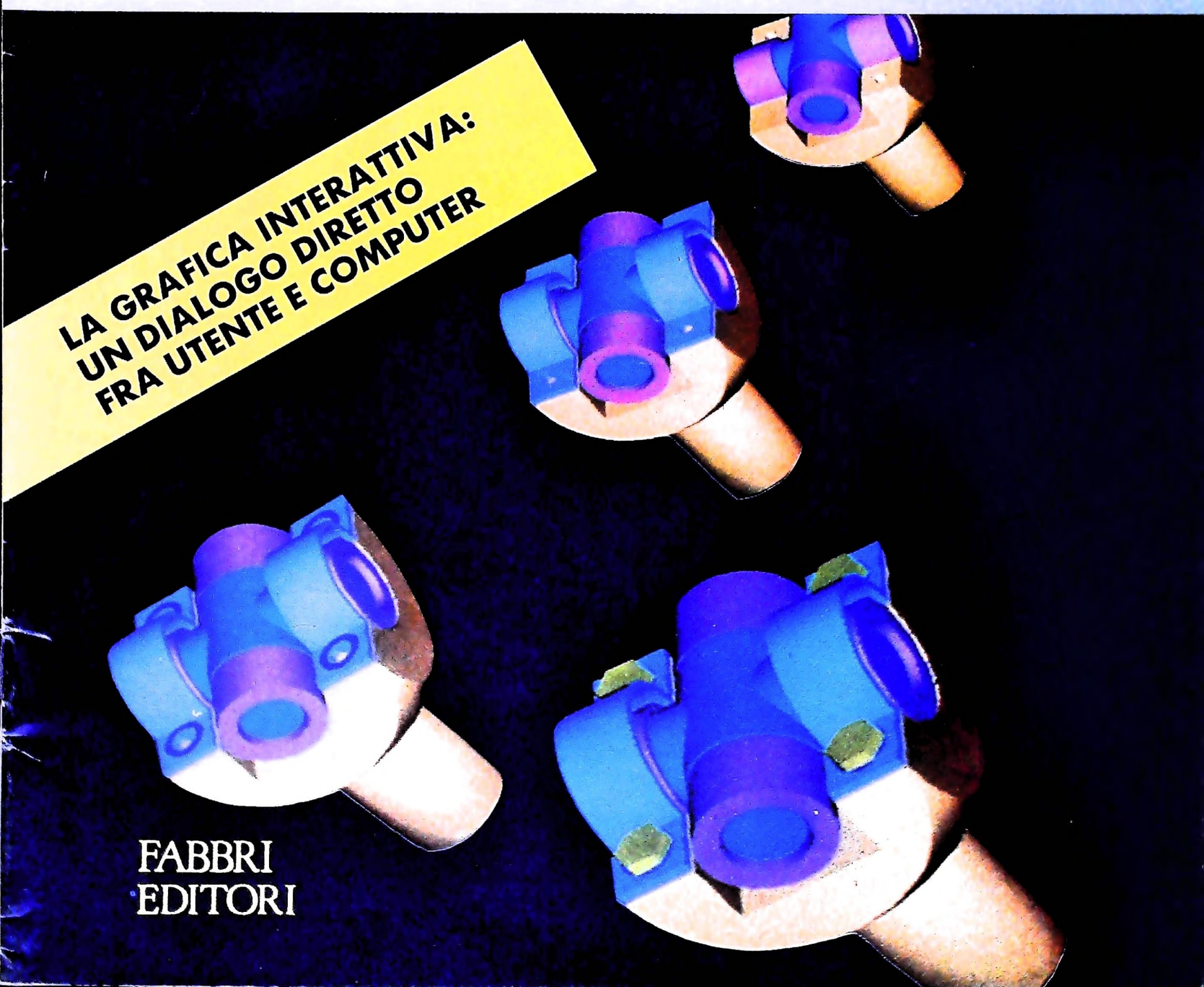
421701

F4 F5 F6 F7 F8

BATTERY LOW.

di GIANNI DEGLI ANTONI

LA GRAFICA INTERATTIVA: UN DIALOGO DIRETTO FRA UTENTE E COMPUTER



FABBRI EDITORI

IL BANCO DI ROMA FINANZIA IL VOSTRO ACQUISTO DI M 10 e M 20

Acquisto per contanti

È la formula di acquisto tradizionale.
Non vi sono particolari commenti da fare, se non sottolineare che troverete ampia disponibilità presso i punti di vendita Olivetti, poiché, grazie al "Corso pratico col computer", godrete di un rapporto di privilegio.

Il servizio di finanziamento bancario

Le seguenti norme descrivono dettagliatamente il servizio di finanziamento offerto dal Banco di Roma e dagli Istituti bancari a esso collegati:

Banca Centro Sud
Banca di Messina
Banco di Perugia

Le agenzie e/o sportelli di questi istituti sono presenti in 216 località italiane.

Come si accede al credito e come si entra in possesso del computer

1) Il Banco di Roma produce una modulistica che è stata distribuita a tutti i punti di vendita dei computer M 10 e M 20 caratterizzati dalla vetrofania M 10.
2) L'accesso al servizio bancario è limitato solo a coloro che si presenteranno al punto di vendita Olivetti.
3) Il punto di vendita Olivetti provvederà a istruire la pratica con la più vicina agenzia del Banco di Roma, a comunicare al cliente entro pochi gorni l'avvenuta concessione del credito e a consegnare il computer.

I valori del credito

Le convenzioni messe a punto con il Banco di Roma, valide anche per le banche collegate, prevedono:
1) Il credito non ha un limite minimo, purché tra le parti acquistate vi sia l'unità computer base.
2) Il valore massimo unitario per il credito è fissato nei seguenti termini:
   — valore massimo unitario per M 10 = L. 3.000.000
   — valore massino unitario per M 20 = L. 15.000.000
3) Il tasso passivo applicato al cliente è pari al "prime rate ABI (Associazione Bancaria Italiana) + 1,5 punti percentuali".
4) La convenzione prevede anche l'adeguamento del tasso passivo applicato al cliente a ogni variazione del "prime rate ABI"; tale adeguamento avverrà fin dal mese successivo a quello a cui è avvenuta la variazione.
5) La capitalizzazione degli interessi è annuale con rate di rimborso costanti, mensili, posticipate; il periodo del prestito è fissato in 18 mesi.
6) Al cliente è richiesto, a titolo di impegno, un deposito cauzionale pari al 10% del valore del prodotto acquistato, IVA inclusa; di tale 10% L. 50.000 saranno trattenute dal Banco di Roma a titolo di rimborso spese per l'istruttoria, il rimanente valore sarà vincolato come deposito fruttifero a un tasso annuo pari all'11%, per tutta la durata del prestito e verrà utilizzato quale rimborso delle ultime rate.
7) Nel caso in cui il cliente acquisti in un momento successivo altre parti del computer (esempio, stampante) con la formula del finanziamento bancario, tale nuovo prestito attiverà un nuovo contratto con gli stessi termini temporali e finanziari del precedente.

Le diverse forme di pagamento del finanziamento bancario

Il pagamento potrà avvenire:
- ☐ presso l'agenzia del Banco di Roma, o Istituti bancari a esso collegati, più vicina al punto di vendita Olivetti;
- ☐ presso qualsiasi altra agenzia del Banco di Roma, o Istituto a esso collegati;
- ☐ presso qualsiasi sportello di qualsiasi Istituto bancario, tramite ordine di bonifico (che potrà essere fatto una volta e avrà valore per tutte le rate);
- ☐ presso qualsiasi Ufficio Postale, tramite vaglia o conto corrente postale. Il numero di conto corrente postale sul quale effettuare il versamento verrà fornito dall'agenzia del Banco di Roma, o da Istituti a esso collegati.

BANCO DI ROMA
CONOSCIAMOCI MEGLIO.

Direttore dell'opera
GIANNI DEGLI ANTONI

Comitato Scientifico
GIANNI DEGLI ANTONI
Docente di Teoria dell'Informazione, Direttore dell'Istituto di Cibernetica dell'Università degli Studi di Milano

UMBERTO ECO
Ordinario di Semiotica presso l'Università di Bologna

MARIO ITALIANI
Ordinario di Teoria e Applicazione delle Macchine Calcolatrici presso l'Istituto di Cibernetica dell'Università degli Studi di Milano

MARCO MAIOCCHI
Professore Incaricato di Teoria e Applicazione delle Macchine Calcolatrici presso l'Istituto di Cibernetica dell'Università degli Studi di Milano

DANIELE MARINI
Ricercatore universitario presso l'Istituto di Cibernetica dell'Università degli Studi di Milano

Curatori di rubriche
TULLIO CHERSI, ADRIANO DE LUCA (Professore di Architettura del Calcolatori all'Università Autonoma Metropolitana di Città del Messico), GOFFREDO HAUS, MARCO MAIOCCHI, DANIELE MARINI, GIANCARLO MAURI, CLAUDIO PARMELLI, ENNIO PROVERA

Testi
ADRIANO DE LUCA, GOFFREDO HAUS, ENNIO PROVERA, Eidos (TIZIANO BRUGNETTI, BRUNO MOTTA, CARMINE STRAGAPEDE)

Tavole
Logical Studio Communication
Il Corso di Programmazione e BASIC è stato realizzato da Etnoteam S.p.A., Milano
Computergrafica è stato realizzato da Eidos, S.c.r.l., Milano
Usare il Computer è stato realizzato in collaborazione con PARSEC S.N.C. - Milano

Direttore Editoriale
ORSOLA FENGHI

Coordinatore settore scientifico
UGO SCAIONI

Redazione
MARINA GIORGETTI
LOGICAL STUDIO COMMUNICATION

Art Director
CESARE BARONI

Impaginazione
BRUNO DE CHECCHI
PAOLA ROZZA

Programmazione Editoriale
ROSANNA ZERBARINI
GIOVANNA BREGGÉ

Segretarie di Redazione
RENATA FRIGOLI
LUCIA MONTANARI

NEL PROSSIMO NUMERO IN OMAGGIO L'OTTAVO POSTER "LA STORIA DELL'INFORMATICA"

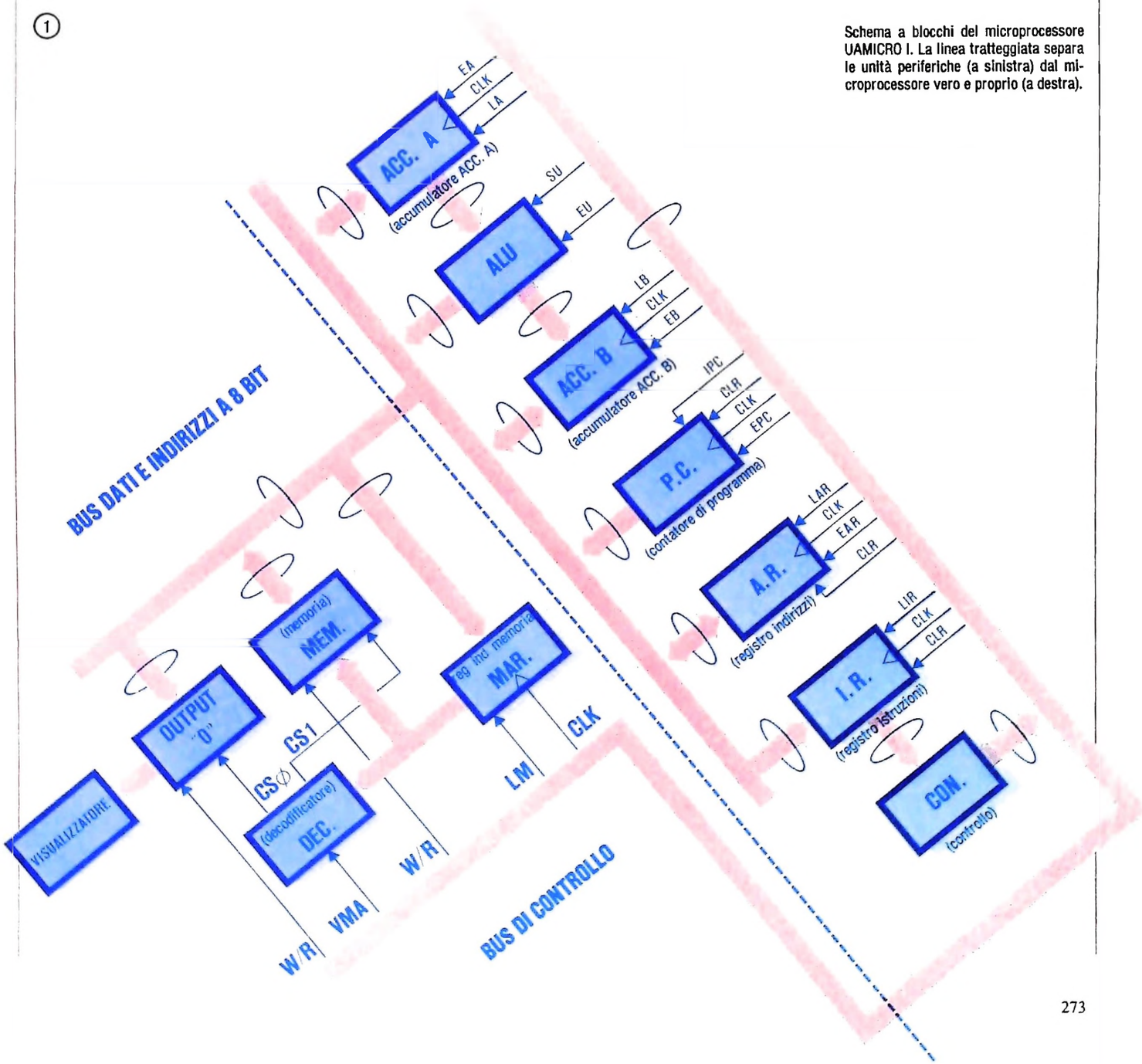
Corso Pratico col Computer - Copyright © sul fascicolo 1984 Gruppo Editoriale Fabbri, Bompiani, Sonzogno, Etas S.p.A., Milano - Copyright © sull'opera 1984 Gruppo Editoriale Fabbri, Bompiani, Sonzogno, Etas S.p.A., Milano - Prima Edizione 1984 - Direttore responsabile GIOVANNI GIOVANNINI - Registrazione presso il Tribunale di Milano n. 135 del 10 marzo 1984. - Iscrizione al Registro Nazionale della Stampa n. 00262, vol. 3, Foglio 489 del 20.9.1982 - Stampato presso lo Stabilimento Grafico del Gruppo Editoriale Fabbri S.p.A., Milano - Diffusione Gruppo Editoriale Fabbri S.p.A. via Mecenate, 91 - tel. 50951 - Milano - Distribuzione per l'Italia: A. & G. Marco s.a.s., via Fortezza 27 - tel. 2526 - Milano - Pubblicazione periodica settimanale - Anno I - n. 18 - esce il giovedì - Spedizione in abb. postale - Gruppo II/70. L'Editore si riserva la facoltà di modificare il prezzo nel corso della pubblicazione, se costretto da mutate condizioni di mercato.

L'UAMICRO I

La prima e più semplice versione del microprocessore UAMICRO.

La figura 1 mostra il microprocessore UAMICRO I. È un microprocessore molto semplice e dalle capacità limitate; esso ha solo lo scopo di illustrare le due grandi funzioni basilari di tutti i calcolatori esistenti oggi: il reperimento dell'istruzione (FETCH) e la sua esecuzione (EXECUTE).
Cominciamo a descrivere le varie parti che lo compongono, partendo dalla linea tratteggiata che lo separa in due zone: a sinistra, le periferiche e gli elementi esterni (memoria, ingresso, visualizzatore ecc.), a destra il microprocessore vero e proprio. Inizieremo la descrizione da questo; è un microprocessore con due BUS: il BUS di controllo formato da 16 linee e il BUS dei dati e indirizzi di 8 linee. Vediamo in particolare cosa vuol dire avere un BUS dei dati e indirizzi di 8 bit; se usiamo gli 8 bit per i dati vuol dire che abbiamo un

①



Schema a blocchi del microprocessore UAMICRO I. La linea tratteggiata separa le unità periferiche (a sinistra) dal microprocessore vero e proprio (a destra).

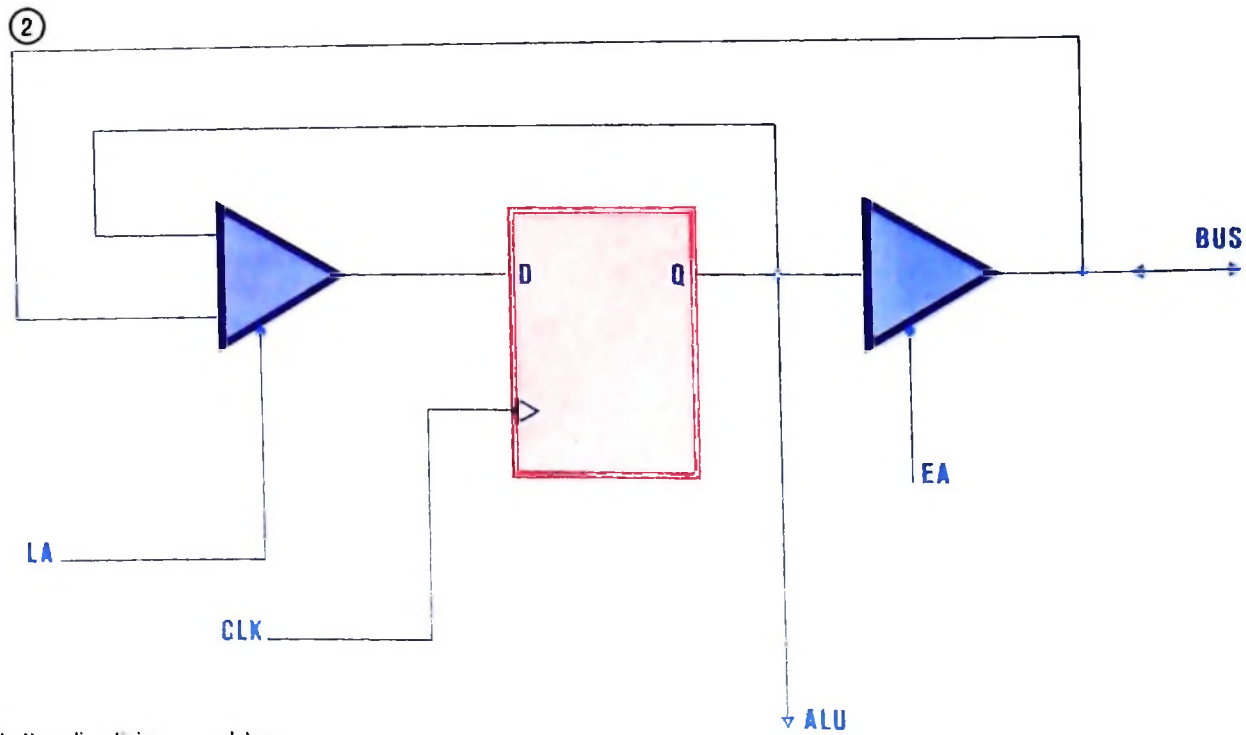
Numerazione binaria ed esadecimale (HEX)

BINARIA	HEX
0 0 0 0 0 0 0 0	0 0
0 0 0 0 0 0 0 1	0 1
0 0 0 0 0 0 1 0	0 2
0 0 0 0 0 0 1 1	0 3
1 1 1 1 1 0 1 0	F A
1 1 1 1 1 0 1 1	F B
1 1 1 1 1 1 0 0	F C
1 1 1 1 1 1 0 1	F D
1 1 1 1 1 1 1 0	F E
1 1 1 1 1 1 1 1	F F

numero che va da 00 a FF nella numerazione esadecimale (vedere la tabella qui a lato). Se li usiamo invece per i codici di istruzioni, avremo a disposizione ben 256 possibili istruzioni diverse, e ciò è eccessivo. In realtà, già con una cinquantina di istruzioni è abbastanza difficile ricordarsele tutte, con duecentocinquanta è impossibile. In base a ricerche statistiche, cui l'autore di questo corso ha partecipato, fatte su una popolazione di circa mille studenti, si è visto chiaramente che solo una trentina di istruzioni erano quelle più usate, le altre molto meno e alcune addirittura mai.
Se usiamo gli otto bit disponibili per gli indirizzi, ciò vuol dire che vi sono 256 (2^8) locazioni di memoria disponibili per i nostri programmi. Esse sono ben poche, ma in questa prima versione non ha importanza la memoria disponibile, né le istruzioni: ciò che conta in questa prima fase è di inquadrare i meccanismi e l'architettura necessaria per realizzare un microprocessore.

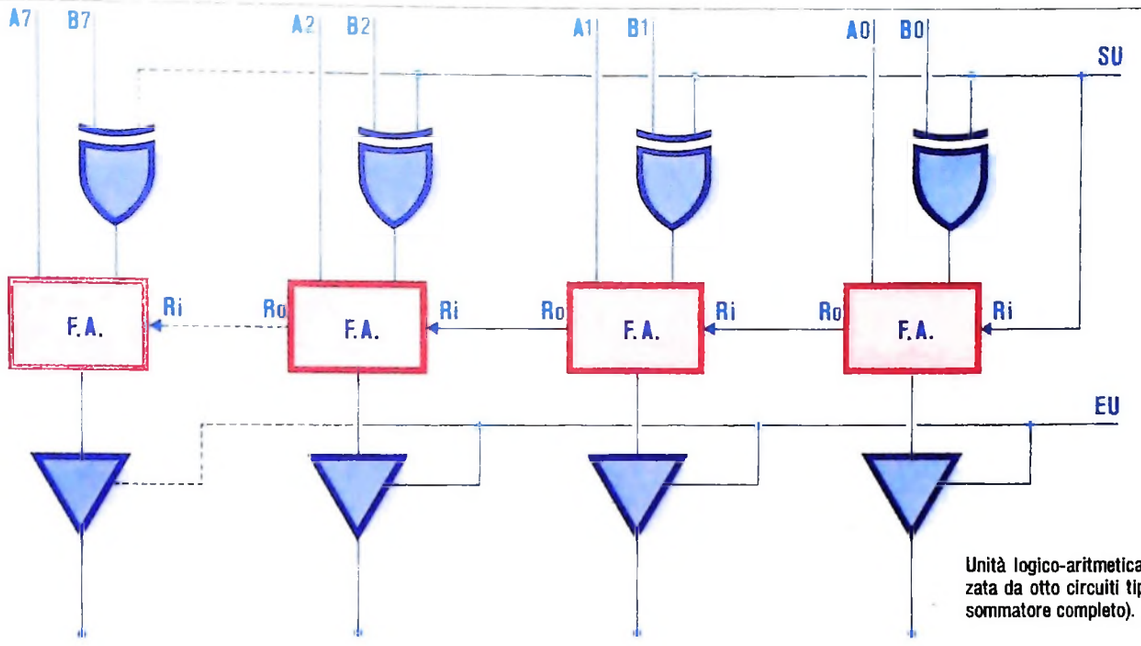
Accumulatori A e B (ACC.A e ACC.B)

I primi registri che incontriamo sono i registri denominati ACC.A e ACC.B (accumulatore A e accumulatore B). Ogni microprocessore ne ha a disposizione almeno uno, ma averne di più è quasi sempre vantaggioso, per cui i microprocessori moderni ne hanno parecchi a disposizione. Sono i registri in cui vengono caricati i dati prima di trasferirli alla memoria oppure agli elementi esterni; oppure, al contrario, vengono trasferiti da questi ultimi verso gli accumulatori per poter essere elaborati internamente al microprocessore.
I due accumulatori comunicano direttamente con l'ALU



Struttura di un tipico accumulatore.

③

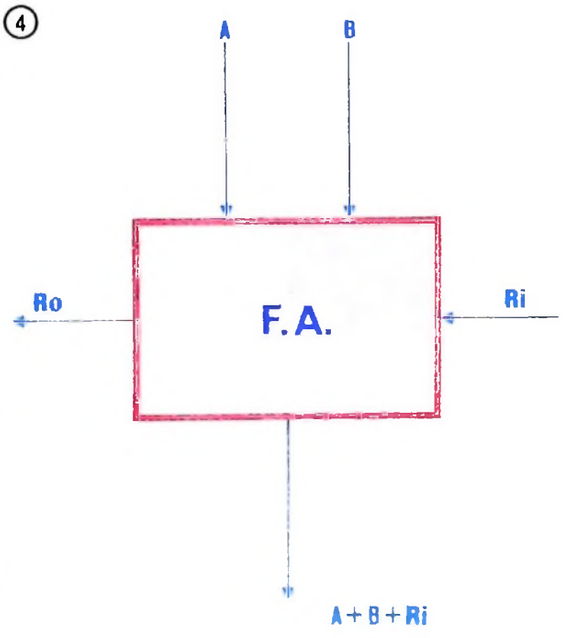


Unità logico-aritmetica (ALU). È realizzata da otto circuiti tipo FA (full adder, sommatore completo).

(unità logico-aritmetica) attraverso due canali monodirezionali, mentre comunicano con il BUS attraverso BUFFER a tre stati, per motivi facilmente comprensibili.
Sono due registri sincroni, per cui uno dei segnali che proviene dal BUS di controllo è il segnale CLK (CLOCK). Inoltre, nell'ACC.A entra il segnale di comando EA (ENABLE A) che ha questa funzione: quando EA = 1 i dati presenti nell'ACC.A passano immediatamente al BUS. Se riprendiamo in esame (figura 2) il circuito visto nella figura 9 a pag. 216 qui già opportunamente collegato al BUS, vediamo che con il comando ENABLE si abilita l'uscita dei dati. Con il comando LA = 1 i dati passano al circuito MULTIPLEXER e alla transizione positiva del CLOCK vengono immagazzinati nel flip-flop. Otto di questi circuiti esistono sia nell'accumulatore A sia in quello B. In questo, i segnali di comando LB ed EB hanno le stesse funzioni degli analoghi dell'accumulatore A.

Struttura di un circuito sommatore completo.

④



ALU (Unità logico-aritmetica)

Questo circuito, realizzabile come mostra la figura 3, è formato, come si può vedere, da otto circuiti F.A. (sommatori completi), da otto OR ESCLUSIVI e da otto BUFFER a tre stati per il collegamento al BUS. Vediamo ora i comandi: con il comando SU = 0 i dati dell'ACC.B passano direttamente assieme a quelli dell'ACC.A al F.A.; essendo questo un circuito asincrono, i due dati vengono immediatamente sommati.
Spieghiamo un po' come funziona un sommatore completo (figura 4); esso è composto da tre entrate e due uscite. Le due entrate A e B sono i due addendi che vengono sommati, l'entrata Ri è il riporto, se c'è, della somma dei due addendi precedenti, mentre l'uscita Ro è il riporto, se c'è, della somma A + B + Ri e, infine, l'uscita A + B + Ri è il risultato della somma.
Così com'è fatto il nostro sommatore, i vari riporti partono da destra e si spostano verso sinistra, dove l'ultimo riporto si perde, nel nostro caso. Nei microprocessori più elaborati, come l'UAMICRO III, questo evento genera un segnale di errore per avvertire che si sono superati i limiti massimi.
Poiché siamo in un ambiente asincrono, i risultati vengono presentati immediatamente all'entrata dei BUFFER a tre stati. Quando invece SU = 1, allora avviene la sottrazione A - B. Vediamo come: poiché, come era già stato detto in precedenza, gli unici circuiti disponibili in un'ALU sono dei som-

matori, per realizzare una sottrazione bisogna usare il metodo del complemento a due. Questo metodo consiste nell'invertire tutti gli "uno" e tutti gli "zero" di un numero binario e poi sommare ad essi un "uno". A esempio:

00101110	numero binario
11010001	inversione
00000001	+ 1 somma
11010010	complemento a 2 del numero di partenza.

Per poter eseguire la somma, basta ricordare le regole dell'aritmetica binaria per cui 1 + 0 = 1, 0 + 0 = 0 e 1 + 1 = 10. Così, quando si somma 1 a 1 si ha 0 più un riporto di 1.

Dopo questa breve spiegazione di come si esegue il complemento a 2, vediamo ora come esso è realizzato nel nostro circuito.
Se poniamo SU = 1 stiamo mettendo un 1 negli OR ESCLUSIVI sia all'ingresso Ri del primo sommatore completo. Quando poniamo un 1 in un ingresso dell'OR ESCLUSIVO, i dati che entrano nell'altro ingresso escono invertiti all'uscita, per cui attraverso gli OR ESCLUSIVI realizziamo la prima operazione, cioè l'inversione; con l'entrata SU = 1 nella Ri del primo circuito realizziamo la seconda operazione, cioè la somma di +1 al risultato dell'inversione. Infine, in questo modo stiamo sommando al primo addendo il complemento a due del secondo, per cui il risultato è la sottrazione di B ad A, cioè A - B.

Glossario

Base di dati - *una raccolta generalizzata e integrata di dati relativi a un'azienda, a un campo del sapere, a una organizzazione ecc., destinata a soddisfare le esigenze di tutte le applicazioni che vi hanno accesso e organizzata in modo da riprodurre le relazioni "naturali" che intercorrono fra i dati in essa archiviati. Una base di dati in generale serve a più applicazioni e ha esistenza e validità indipendenti dai singoli programmi applicativi. Con il diffondersi dei piccoli calcolatori, il termine e il suo equivalente inglese (data base) hanno assunto il significato più generico di qualunque raccolta di dati relativamente omogenei fra loro, memorizzati su supporto magnetico in modo da renderne agevole il recupero selettivo in funzione delle esigenze dell'utente.*

Base di dati distribuita - *in un sistema informativo distribuito, base di dati unica da un punto di vista logico, ma suddivisa fisicamente fra più sedi.*

Concentratore - *apparecchiatura che consente di far confluire i dati provenienti da un certo numero di collegamenti o linee di trasmissione in un minor numero di collegamenti a maggiore capacità e di smistare il flusso inverso dalle linee o collegamenti a grande capacità sui collegamenti minori.*

EOM (End of Message) - *fine del messaggio. Nelle comunicazioni, codice di controllo internazionale che indica la conclusione della trasmissione di un testo oppure di più testi, che formano un messaggio completo.*

EOT (End of Transmission) - *fine della trasmissione. Nelle comunicazioni, codice di controllo internazionale che indica la conclusione di una sequenza di trasmissione e riporta in condizione di riposo i terminali o le apparecchiature interessate.*

EXT (End of Text) - *fine del testo. Nelle comunicazioni, codice di controllo di trasmissione utilizzato per indicare la conclusione di un testo.*

Filtro - *dispositivo per il controllo delle frequenze che possono passare in un circuito: esso consente alle frequenze selezionate di transitare senza attenuazione significativa e blocca le frequenze che sono indesiderate.*

Filtro a reiezione di banda - *filtro che consente il passaggio in un circuito di tutte le frequenze al di sotto e al di sopra di una banda specifica: le frequenze all'interno di tale banda sono invece bloccate.*

Filtro passa alto - *filtro che consente il passaggio in un circuito di tutte le frequenze al di sopra di un determinato valore di soglia (frequenza di taglio), mentre blocca il passaggio a tutte le frequenze inferiori.*

Filtro passa banda - *filtro che consente il passaggio in un circuito di una banda di frequenze e blocca il passaggio di tutte le frequenze al di sotto e al di sopra della banda passante.*

Filtro passa basso - *filtro che consente il passaggio in un circuito di tutte le frequenze al di sotto di un determinato valore di soglia (frequenza di taglio), mentre blocca il passaggio a tutte le frequenze superiori.*

Hertz - *unità di misura della frequenza equivalente a un ciclo al secondo: è la frequenza di un fenomeno periodico il cui periodo è pari a 1 secondo. Una corda in vibrazione, per esempio, produce un suono di 400 hertz se compie 400 oscillazioni complete in un secondo. Questa unità di misura prende il nome da Heinrich Hertz, scienziato tedesco che contribuì alla scoperta della natura delle onde elettromagnetiche. Il simbolo relativo è Hz; i multipli più usati sono il kilohertz (kHz, 1000 hertz), il megahertz (MHz, un milione di hertz) e il gigahertz (GHz, un miliardo di hertz).*

Multiplazione o Multiplexing - *realizzazione di due o più collegamenti logici mediante un unico collegamento fisico.*

Multiplatore o Multiplexer - *apparecchiatura che realizza la funzione di multiplazione. La funzione è simile a quella di concentrazione ma, a differenza di un concentratore, un multiplatore non è mai programmabile.*

Ospite (Host) - *nell'elaborazione distribuita, un calcolatore che fornisce servizi o svolge funzioni di controllo nei confronti degli utilizzatori o dei calcolatori satellite, dei terminali e delle altre apparecchiature ausiliarie. Un calcolatore ospite in genere è considerato autosufficiente e non richiede supervisione da parte di altri calcolatori.*

IL MODO MINORE

Come per il modo maggiore, esaminiamo anche per il modo minore le caratteristiche delle regioni.

Parlando del modo maggiore abbiamo introdotto il concetto di *regione*, cioè di tonalità allargata alle scale delle tonalità più affini, in termini di relazioni armoniche. Anche nel caso del modo minore possiamo parlare di regioni, anche se il modo minore esercita un controllo meno diretto sulle altre tonalità. Infatti, mentre nel modo maggiore la *tonica* esercita un ruolo di *dominante* per la sua *sottodominante* (ad esempio, in *do maggiore* la triade maggiore *do-mi-sol* è sette semitoni, cioè una quinta, sopra la triade maggiore *fa-la-do* che a sua volta è cinque semitoni, cioè una quarta, sopra il *do*), nel modo minore questa relazione non vale più; ad esempio, in *la minore* la tonica (cioè la triade minore *la-do-mi*) non può più esercitare la funzione di dominante rispetto alla triade *re-fa-la* (sottodominante della tonica) poiché una triade dominante deve essere maggiore (cioè con intervalli tra le altezze che la costituiscono *4-3*) allo stesso modo della triade *mi-sol diesis-si*.

Le regioni più vicine rispetto ad una certa tonica nel modo minore sono la *mediante maggiore*, la *dominante minore*, la *tonica maggiore* e la *sottodominante minore*; ad esempio, per la tonalità di *la minore*, le regioni più vicine sono il *do maggiore*, il *mi minore*, il *la maggiore* e il *re minore*. Le modulazioni in un brano in modo minore saranno preferenzialmente verso queste tonalità affini.

La modulazione a una tonalità più lontana è comunque molto usata, tuttavia essa richiede una *preparazione* maggiormente complessa, cioè tipicamente una sequenza di modulazioni intermedie di passaggio.

Un programma per la generazione di sequenze di note in "la minore"

In questo programma vedremo come generare sequenze di note in *la minore* mediante la generazione di numeri pseudo-casuali. Il programma è strutturato in cinque parti principali: inizializzazione e dimensionamento dei vettori; caricamento dei vettori; interazione musicista-programma; generazione delle note; subroutine.

Nella prima parte sono definiti i vettori arm, mel, zin e nat destinati a memorizzare le scale del modo minore (armonica, melodica, zingaresca e naturale, rispettivamente); sono vettori costituiti da nove elementi ciascuno in corrispondenza con i sette gradi ascendenti (elementi 0-6), il settimo (7) ed il sesto (8) grado discendenti.

Nel blocco di istruzioni 069-103 carichiamo i vettori con le scale. Nel blocco 031-042 memorizziamo invece la struttura ritmica, riprodotta qui in basso, che useremo nel programma ciclicamente per mantenere una base ritmica costante; naturalmente potremmo memorizzarne un'altra (più breve, più lunga, strutturalmente differente) o memorizzarne più d'una ed usarle poi in alternativa in dipendenza da una determinata condizione (per esempio, in dipendenza dal valore del numero pseudo-casuale attuale).

Nella parte di interazione scegliamo il tempo (tra 0 e 5, dal lento al veloce), la scala di riferimento per la generazione delle altezze (1 per la scala armonica, 2 per la scala melodica, 3 per la scala naturale, 4 per la scala zingaresca) e infine

Memorizzazione della struttura ritmica nel blocco 031 - 042, usata nel programma ciclicamente per mantenere una base ritmica costante.

diamo un numero intero in dipendenza dal quale il generatore pseudo-casuale produrrà sequenze differenti.
Poiché il generatore pseudo-casuale produce numeri reali compresi tra 0 e 1 moltiplichiamo per 10 il valore generato: usiamo quindi la sua parte intera per indirizzare all'interno dei vettori che contengono le scale minori. Inoltre, quando il numero generato è compreso tra 0 e 0.33 passiamo all'ottava superiore all'attuale, se è compreso tra 0.66 e 1 passiamo all'ottava inferiore all'attuale; i salti d'ottava li eseguiamo solo se generiamo gradi della scala minore diversi dal sesto e dal settimo ascendenti o discendenti. Controlliamo naturalmente di non uscire dal range frequenziale dell'M10 con le operazioni di salto d'ottava (istruzioni 215-217).
Con la subroutine 5000-5530 scandiamo ciclicamente il vettore f in cui abbiamo memorizzato la struttura ritmica calcolando il valore di durata in base all'indicazione di tempo comunicata al momento del lancio del programma. Una tecnica molto usata per l'affinamento delle composizioni automatiche, e che qui vogliamo solo esemplificare, consiste nel richiamare, in dipendenza da qualche parametro, un tema melodico ricorrente (eventualmente trasformato in qualche modo di volta in volta).
Nel nostro caso, nel blocco 6000-6009 viene caricato il tema ricorrente nel vettore tem, quindi nel blocco di istruzioni 6015-6050 viene generata la sequenza del tema melodico. Questa subroutine viene richiamata quando la variabile i assume valori superiori a 8 e di conseguenza non è utilizzabile per indirizzare all'interno dei vettori in cui sono memorizzate le scale minori.
Per semplicità abbiamo caricato staticamente una scala melodica completa ascendente e discendente; pertanto verrà sempre eseguita questa scala inalterata ogni volta che i supera il valore 8; ma una prova interessante è mettere nel blocco 6000-6009 istruzioni che modificano il tema ricorrente e che quindi portano alla esecuzione di volta in volta di un tema differente (anche se dipendente dal tema eseguito la volta precedente). Per questo il blocco di caricamento è posizionato all'interno della subroutine 5999-6050 piuttosto che all'inizio del programma: è un blocco inteso in senso generale come blocco di trasformazione di un tema ricorrente; nel caso di un tema statico è chiaramente più conveniente caricarlo una sola volta all'inizio del programma piuttosto che ogni volta che viene richiamata la subroutine. Da notare che il trattamento del parametro durata all'interno di questa subroutine mantiene la struttura ritmica di base memorizzata nel vettore f.

Programma per generare sequenze di note in la minore

```
010 clear
015 dim f(11),tem(13)
020 REM programma per la
    generazione di note in modo
    minore
025 REM memorizzazione scale minori
030 dim arm(8),mel(8),zin(8),nat(8)
031 f(0)=128
032 f(1)=64
033 f(2)=64
034 f(3)=192
035 f(4)=32
036 f(5)=32
037 f(0+6)=128
038 f(1+6)=128
039 f(2+6)=64
040 f(3+6)=64
041 f(4+6)=64
042 f(5+6)=64
069 REM scala armonica (la minore)
070 arm(0)=11172
071 arm(1)=9952
072 arm(2)=9394
073 arm(3)=8368
074 arm(4)=7456
075 arm(5)=7032
076 arm(6)=5918
077 arm(7)=arm(6)
078 arm(8)=arm(5)
079 REM scala melodica
080 for z=0 to 8
081 mel(z)=arm(z)
082 next z
083 mel(5)=6642
084 mel(7)=6269
089 REM scala naturale
090 for z=0 to 8
091 nat(z)=arm(z)
092 next z
093 nat(6)=6269
094 nat(7)=nat(6)
099 REM scala zingaresca
100 for z=0 to 8
101 zin(z)=arm(z)
102 next z
103 zin(3)=7900
109 REM interazione
110 input "tempo (da 0 a 5)";tmp
115 if tmp>5 or tmp<0 then goto 110
122 j=0
130 input "scegli la scala
    (da 1 a 4)";sc
140 if sc<>1 and sc<>2 and sc<>3
    and sc<>4 goto 130
145 oldi=0
150 REM generazione pseudocasuale
160 input "scrivi un numero
    intero";INIT
170 rn=rnd(INIT)
172 defint i
```

diamo un numero intero in dipendenza dal quale il generatore pseudo-casuale produrrà sequenze differenti.
Poiché il generatore pseudo-casuale produce numeri reali compresi tra 0 e 1 moltiplichiamo per 10 il valore generato: usiamo quindi la sua parte intera per indirizzare all'interno dei vettori che contengono le scale minori. Inoltre, quando il numero generato è compreso tra 0 e 0.33 passiamo all'ottava superiore all'attuale, se è compreso tra 0.66 e 1 passiamo all'ottava inferiore all'attuale; i salti d'ottava li eseguiamo solo se generiamo gradi della scala minore diversi dal sesto e dal settimo ascendenti o discendenti. Controlliamo naturalmente di non uscire dal range frequenziale dell'M10 con le operazioni di salto d'ottava (istruzioni 215-217).
Con la subroutine 5000-5530 scandiamo ciclicamente il vettore f in cui abbiamo memorizzato la struttura ritmica calcolando il valore di durata in base all'indicazione di tempo comunicata al momento del lancio del programma. Una tecnica molto usata per l'affinamento delle composizioni automatiche, e che qui vogliamo solo esemplificare, consiste nel richiamare, in dipendenza da qualche parametro, un tema melodico ricorrente (eventualmente trasformato in qualche modo di volta in volta).
Nel nostro caso, nel blocco 6000-6009 viene caricato il tema ricorrente nel vettore tem, quindi nel blocco di istruzioni 6015-6050 viene generata la sequenza del tema melodico. Questa subroutine viene richiamata quando la variabile i assume valori superiori a 8 e di conseguenza non è utilizzabile per indirizzare all'interno dei vettori in cui sono memorizzate le scale minori.
Per semplicità abbiamo caricato staticamente una scala melodica completa ascendente e discendente; pertanto verrà sempre eseguita questa scala inalterata ogni volta che i supera il valore 8; ma una prova interessante è mettere nel blocco 6000-6009 istruzioni che modificano il tema ricorrente e che quindi portano alla esecuzione di volta in volta di un tema differente (anche se dipendente dal tema eseguito la volta precedente). Per questo il blocco di caricamento è posizionato all'interno della subroutine 5999-6050 piuttosto che all'inizio del programma: è un blocco inteso in senso generale come blocco di trasformazione di un tema ricorrente; nel caso di un tema statico è chiaramente più conveniente caricarlo una sola volta all'inizio del programma piuttosto che ogni volta che viene richiamata la subroutine. Da notare che il trattamento del parametro durata all'interno di questa subroutine mantiene la struttura ritmica di base memorizzata nel vettore f.

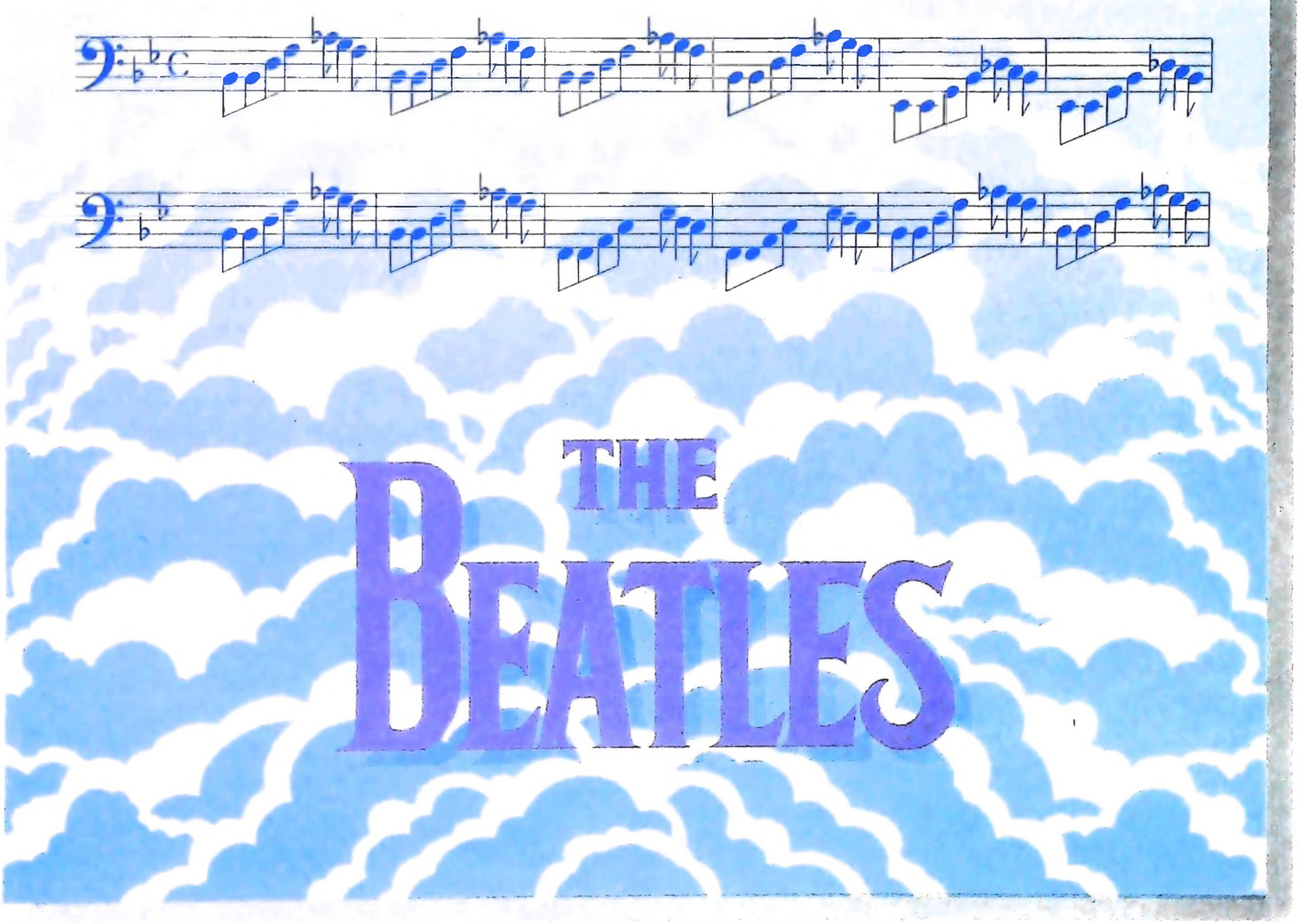
Programma per generare sequenze di note in la minore

```
010 clear
015 dim f(11),tem(13)
020 REM programma per la
    generazione di note in modo
    minore
025 REM memorizzazione scale minori
030 dim arm(8),mel(8),zin(8),nat(8)
031 f(0)=128
032 f(1)=64
033 f(2)=64
034 f(3)=192
035 f(4)=32
036 f(5)=32
037 f(0+6)=128
038 f(1+6)=128
039 f(2+6)=64
040 f(3+6)=64
041 f(4+6)=64
042 f(5+6)=64
069 REM scala armonica (la minore)
070 arm(0)=11172
071 arm(1)=9952
072 arm(2)=9394
073 arm(3)=8368
074 arm(4)=7456
075 arm(5)=7032
076 arm(6)=5918
077 arm(7)=arm(6)
078 arm(8)=arm(5)
079 REM scala melodica
080 for z=0 to 8
081 mel(z)=arm(z)
082 next z
083 mel(5)=6642
084 mel(7)=6269
089 REM scala naturale
090 for z=0 to 8
091 nat(z)=arm(z)
092 next z
093 nat(6)=6269
094 nat(7)=nat(6)
099 REM scala zingaresca
100 for z=0 to 8
101 zin(z)=arm(z)
102 next z
103 zin(3)=7900
109 REM interazione
110 input "tempo (da 0 a 5)";tmp
115 if tmp>5 or tmp<0 then goto 110
122 j=0
130 input "scegli la scala
    (da 1 a 4)";sc
140 if sc<>1 and sc<>2 and sc<>3
    and sc<>4 goto 130
145 oldi=0
150 REM generazione pseudocasuale
160 input "scrivi un numero
    intero";INIT
170 rn=rnd(INIT)
172 defint i
```

```
175 i=rn*10
177 REM richiamo tema ricorrente
178 if i>8 then gosub 5000
179 REM richiamo subroutines scale
    minori
180 on sc goto 1000,2000,3000,4000
200 REM salti d'ottava
212 if rn<0.33 and i<>5 and i<>6
    and i<>7 and i<>8 then x=x/2
214 if rn>0.66 and i<>5 and i<>6
    and i<>7 and i<>8 then x=x*2
215 REM normalizzazione nel range
    frequenziale dell'M10
216 if x<200 then x=x*2
217 if x>16383 then x=x/2
218 REM generazione delle durate
219 gosub 5000
220 REM generazione di una nota
221 sound x,y
230 rn=rnd(rn)
240 goto 175
1000 REM sequenze scala armonica
```

Esercizio: i cicli in "Birthday"

Proviamo ora ad affrontare un secondo problema di realizzazione musicale sul personal computer M10. Consideriamo la parte di basso all'inizio della canzone dei Beatles "Birthday".
Invece di codificare sequenzialmente le note presenti sul pentagramma (mediante l'istruzione SOUND), sarà interessante trovare un modo di descrivere questo testo musicale usando le istruzioni che permettono di realizzare cicli, sia rispetto al parametro di altezza che al parametro durata.
Proviamo! Prossimamente vedremo insieme qualche soluzione tra le infinite possibili.



```
1020 x=arm(i)
1030 goto 212
2000 REM sequenze scala melodica
2010 if oldi=5 then i=6
2020 if oldi=6 then i=0
2030 if oldi=7 then i=8
2040 if oldi=8 then i=4
2050 x=mel(i)
2060 oldi=i
2070 goto 212
3000 REM sequenze scala naturale
3010 x=nat(i)
3020 goto 212
4000 REM sequenze scala zingaresca
4010 x=zin(i)
4020 goto 212
5000 REM struttura ritmo
5300 y=f(j)/2^tmp
5500 j=j+1
5520 if j=12 then j=0
5530 return
5999 REM tema ricorrente
```

```
6000 REM caricamento del tema
6001 for w=0 to 6
6002 tem(w)=mel(w)
6003 next w
6004 tem(7)=5586
6005 tem(8)=mel(7)
6006 tem(9)=mel(8)
6007 for w=10 to 13
6008 tem(w)=mel(14-w)
6009 next w
6015 REM esecuzione del tema
6016 for w=0 to 13
6018 x=tem(w)
6020 y=f(j)/2^tmp
6030 j=j+1
6040 if j=12 then j=0
6043 sound x,y

6046 next w
6048 REM normalizzazione indice
6049 i=i/2
6050 return
```

Fairlight CMI

Il Fairlight CMI (Computer Musical Instrument) è uno fra i più raffinati strumenti musicali a calcolatore: la configurazione standard, come si può vedere nell'illustrazione, comprende una tastiera musicale (in apparenza non dissimile da quella di un più tradizionale organo elettronico), una tastiera da calcolatore per il controllo e l'immissione di dati, un'unità video dotata di penna ottica, un'unità centrale (visibile qui sul fondo) che contiene un vero e proprio computer e due unità di lettura per dischetti flessibili. Il Fairlight consente di campionare suoni naturali e acustici mediante microfono o nastro e di riprodurli quindi, controllandoli da tastiera; consente anche di sintetizzare suoni per via completamente elettronica e di miscelare suoni di sintesi con suoni naturali campionati. Può essere suonato in tempo reale con la tastiera, come un sintetizzatore normale, ma può anche memorizzare un brano ed eseguirlo automaticamente: sono memorizzabili fino a 50.000 note in un'unica composizione su dischetto. Il sistema può essere pilotato anche da una chitarra e programmi appositi consentono la stampa di musica, su apposita stampante grafica.

Lezione 17

I tipi "user defined" in Basic

Abbiamo già visto che il BASIC mette a disposizione solo tipi numerici e il tipo stringa e non consente all'utente di definire nuovi tipi. Come comportarsi in tutte le situazioni in cui i valori da elaborare non sono né numeri né stringhe? Non resta altro che individuare un modello di rappresentazione che associ a ciascun valore dell'utente in modo univoco un numero. Così per rappresentare il tipo colore definito dai valori rosso, blu, verde, giallo, nero potremo attribuire a ciascuno un numero crescente partendo per esempio da 0 e associando così il rosso allo 0, il blu all'1, il verde al 2 e così via. Come promemoria e per rendere il programma comprensibile anche a chi non avesse partecipato alla sua costruzione, mostreremo con opportuni commenti la corrispondenza tra i valori numerici e quelli rappresentati. Nel caso dei programmi che usano variabili booleane potremo scrivere così:

```
10 'TYPE SWITCH=BOOLEAN: 1=vero; 0=falso
20 'IF SWITCH THEN ...
30 IF SWITCH=1 THEN ...
```

Il programma per il listino prezzi risulterebbe invece così:

```
10 DIM PREZZI(5)
20 ' I valori da 1 a 5 sono associati ai seguen
ti colori:
30 ' rosso,blu,verde,giallo,nero
32 FOR I=1 TO 5
34 READ PREZZO(I)
36 NEXT I
40 PRINT"    INFORMATORE PREZZI VERNICI"
50 PRINT
60 PRINT"Colori disponibili:"
70 PRINT"       ROSSO  :1"
80 PRINT"       BLU    :2"
90 PRINT"       VERDE  :3"
100 PRINT"       GIALLO :4"
110 PRINT"       NERO   :5"
120 INPUT "Colore richiesto";I
130 IF I>=1 AND I<=5 GOTO 160
140 PRINT"ERRORE"
150 GOTO 120
160 PRINT"Prezzo: ";PREZZO(I)
165 DATA 10000,12000,8000,15000,12000
170 ' Fine programma
```

Completata questa diciassettesima lezione del Corso di Programmazione e BASIC, siete in grado di eseguire gli esercizi

MONTET.DO
MONTEP. BA
INTET. DO
INTEP. BA
DADIT. DO
DADIP. BA

contenuti nella cassetta "5 esercizi di programmazione", lato B.

I titoli seguiti dal suffisso DO corrispondono a testi, quelli seguiti da BA a programmi in BASIC.
Caricateli secondo le modalità che avete appreso.

La rappresentazione dei caratteri.

Un calcolatore può rappresentare l'informazione solo attraverso due tipi di "simboli", che corrispondono al fatto che i suoi circuiti si comportano come un interruttore aperto oppure chiuso.
Solitamente si indicano simbolicamente tali configurazioni con i simboli "0" e "1".
Da qui è possibile comporre tali simboli per rappresentare

- valori numerici
- valori di stringhe
- valori "logici" (vero o falso)

e tutto ciò che riteniamo rilevante.
In particolare, i caratteri sono rappresentati con valori che, qualora fossero interpretati come numeri, risulterebbero ordinati proprio come nell'ordine alfabetico.
Possiamo verificarlo.

In BASIC c'è a disposizione la funzione ASC che, applicata a una stringa, ne fornisce la sua rappresentazione interpretata come valore numerico.

Così

```
10 PRINT ASC("A");ASC("B");ASC("C")
```

dà come risultato:

```
65  66  67
```

Osserviamo anche la differenza che c'è tra il valore 4 e il carattere '4'!

```
10 PRINT 4,ASC("4")
```

che, eseguito, fornisce il seguente risultato:

```
4                 52
```

È disponibile anche la funzione inversa, che fa passare da un valore numerico al carattere corrispondente:

```
10 PRINT CHR$(65);CHR$(66);CHR$(67)
```

che fornisce la stampa di "A", di "B" e di "C" nel modo seguente:

```
ABC
```

Il seguente programma visualizza tutti i caratteri disponibili:

```
10 FOR I=0 TO 255
20 PRINT CHR$(I)
30 NEXT I
```

Si noti l'importanza dei commenti inseriti per consentire la comprensione della funzione logica effettuata dal programma anche a chi non lo ha scritto. Si noti inoltre che la presente realizzazione, che usa i valori del tipo colore simulato per accedere all'elemento cercato nella tabella, consente di evitare la presenza di un secondo array che contenga i colori delle vernici e rende il reperimento dell'elemento cercato molto più efficiente.
Infatti procedendo in questo modo, l'accesso all'elemento è immediato, mentre nel caso della doppia tabella, prezzi e colori, sarebbe stata necessaria una scansione della tabella dei colori per individuare la posizione cercata.
Prendiamo ora in considerazione il caso del programma che calcola l'area delle figure geometriche:

```
10 PRINT "CALCOLO AREE FIGURE GEOMETRICHE"
20 PRINT
30 PRINT "       quadrato     : 1"
40 PRINT "       rettangolo   : 2"
50 PRINT "       trapezio     : 3"
60 PRINT "       pentagono    : 4"
70 PRINT "       cerchio      : 5"
80 INPUT "Quale figura";F
90 IF F<1 OR F>5 GOTO 80
100 ON F GOTO 200,300,400,500,600
200 'Calcolo area quadrato
210 PRINT"CALCOLO AREA QUADRATO"
220 GOTO 700
300 ' Calcolo area rettangolo
310 PRINT"CALCOLO AREA RETTANGOLO"
320 GOTO 700
400 'Calcolo area trapezio
410 PRINT"CALCOLO AREA TRAPEZIO"
420 GOTO 700
500 'Calcolo area pentagono
510 PRINT"CALCOLO AREA PENTAGONO"
520 GOTO 700
600 'Calcolo area cerchio
610 PRINT"CALCOLO AREA CERCHIO"
620 GOTO 700
700 'Fine programma
```

(il programma è da completare e le istruzioni per il calcolo delle aree sono indicate dai corrispondenti commenti); si noti in particolare l'uso di una nuova istruzione, la ON...GOTO che consente di trasferire il controllo al numero di linea opportuno sulla base del valore della variabile indicata.
Così, se la variabile F assume il valore 1, il controllo viene trasferito al numero di linea 200, se la variabile assume il valore 2 il controllo viene trasferito all'istruzione 300 e così via di seguito.

Il concetto di tipo subrange

In alcuni casi una variabile può assumere solo un sottoinsieme dei valori di un certo tipo. Facciamo un esempio:
una variabile che rappresenti i mesi dell'anno assume valori interi, ma solo un sottoinsieme di quelli disponibili. Non sarebbe infatti significativo per una tale variabile, per esempio, il valore 13. Allo stesso modo se immaginiamo di aver definito il tipo GIORNIDELLASETTIMANA nel modo seguente:

```
TYPE GIORNIDELLASETTIMANA = (LUNEDÌ,MARTEDÌ,MERCOLEDÌ,GIOVEDÌ,VENERDÌ, SABATO,DOMENICA)
```

un programma che visualizzi i turni di lavoro nei giorni lavorativi userebbe soltanto i valori da LUNEDÌ a VENERDÌ.
I linguaggi di programmazione avanzati come il PASCAL consentono di definire tipi di dati costruiti a partire da tipi già definiti. Per esempio, nei casi suddetti sarebbe possibile definire il tipo MESE nel modo seguente:

```
TYPE MESE = 1...12
```

e il tipo GIORNILAVORATIVI nel modo seguente:

```
TYPE GIORNILAVORATIVI=LUNEDÌ...VENERDÌ
```

Si dice in questo caso che i tipi MESE e GIORNILAVORATIVI sono definiti per "subrange", che in inglese significa appunto "sottoinsieme". Le definizioni sopra date significano infatti che le variabili di tipo MESE possono assumere solo i valori interi compresi tra 1 e 12 e che le variabili di tipo GIORNILAVORATIVI possono assumere solo i valori LUNEDÌ, MARTEDÌ, ...fino a VENERDÌ. Come si vede è possibile definire tipi subrange sia partendo da tipi già presenti nel linguaggio, sia partendo da tipi definiti dall'utente.

Cosa abbiamo imparato

In questa lezione abbiamo visto:

- La realizzazione di tipi definiti dall'utente in BASIC
- L'istruzione BASIC ON... GOTO
- Il concetto di tipo subrange e la relativa definizione in PASCAL
- La rappresentazione interna al calcolatore di caratteri e numeri
- Le funzioni BASIC ASC e CHR$

INFORMATICA E FISICA

Esaminiamo l'uso dell'elaboratore nell'insegnamento di un'altra materia scientifica.

La didattica della fisica è un campo abbastanza vasto nel quale può essere impiegato utilmente l'elaboratore. Sono state fatte molte ricerche e sperimentazioni e attualmente esistono parecchi programmi a disposizione dell'utente. Le metodologie sono di tipo diverso; risultano certamente predominanti la metodologia tutoriale e quella delle esercitazioni.
Molto più interessante, anche dal punto di vista della ricerca di nuove strade, è la metodologia della simulazione che, come si è già visto in un articolo precedente, mette a disposizione dell'utente tutta la potenza di calcolo e la capacità grafica dell'elaboratore, per riprodurre esperimenti che altrimenti risulterebbero lunghi, complessi, pericolosi o addirittura impossibili ad essere eseguiti.
Vogliamo presentare un esempio significativo e in certo senso rappresentativo della categoria "simulazione". È il programma KEPLERO per la simulazione di orbite planetarie; un secondo programma il FARADAY per la visualizzazione delle linee di campo tra due cariche elettrostatiche, viene accennato con i risultati che appaiono sullo schermo. Entrambi sono stati scritti per l'elaboratore M20 tipo ST della Olivetti.

Il programma KEPLERO per la simulazione di orbite planetarie

Di questo programma non scriviamo la struttura generale perché essendo lineare, cioè senza ramificazioni, si presta ad essere illustrato direttamente.
Si tratta della simulazione di un sistema fisico costituito da una massa centrale, praticamente fissa perché di gran lunga maggiore della massa dei corpi orbitali, i quali, a seconda della posizione e della velocità, descrivono orbite ellittiche oppure iperboliche. Le leggi del moto sono quelle della meccanica newtoniana per il moto in due dimensioni, e precisamente:

posizione

$$x = x_0 + V_x t + 1/2\, a_x t^2$$

$$y = y_0 + V_y t + 1/2\, a_y t^2$$

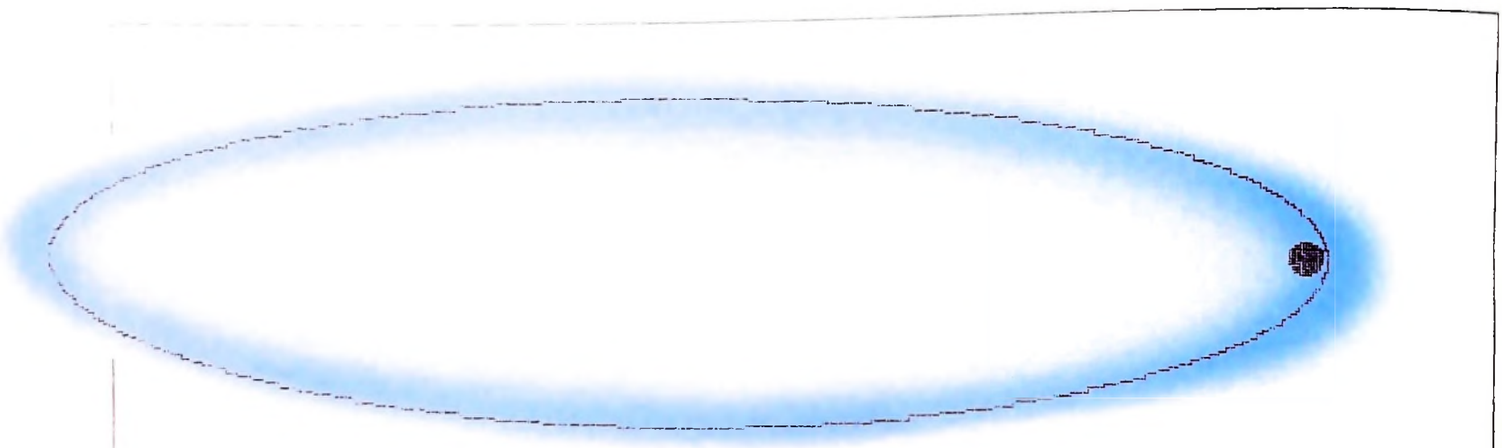
velocità

$$V_x = V_{0x} + a_x t$$

$$V_y = V_{0y} + a_y t$$

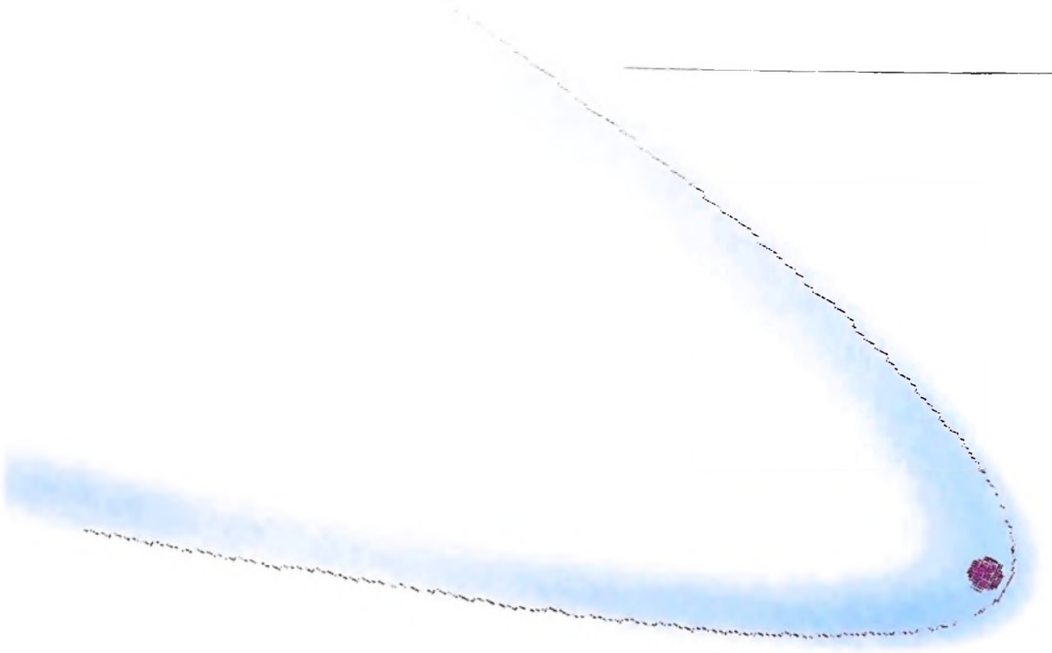
Rileviamo che per la descrizione del moto sono sufficienti due dimensioni, perché, anche se si svolge nello spazio a tre dimensioni, si tratta comunque di un moto piano.
Facciamo notare anzitutto che, nel calcolare la posizione, i termini $1/2\, a_x t^2$ e $1/2\, a_y t^2$ possono essere trascurati perché molto piccoli rispetto agli altri, che compaiono nelle rispettive espressioni.
Per il funzionamento del sistema è necessario che l'utente stabilisca il valore della massa centrale in chilogrammi; la relativa posizione è fissata automaticamente nell'origine di un sistema di assi cartesiani. Occorre quindi stabilire il valore iniziale della posizione e della velocità del corpo orbitante e un intervallo di tempo (DT) tra due posizioni successive. L'elaboratore calcola quindi la posizione negli istanti seguenti, fino al raggiungimento di un'orbita completa oppure di una porzione di traiettoria voluta.
Nella sua struttura generale il programma calcola e memorizza le coordinate di un certo numero di posizioni (fino ad un massimo di 2000) quindi passa in rassegna i relativi valori per determinarne gli estremi e fissare la scala in modo da contenere tutti i punti, infine visualizza sullo schermo la traiettoria.

```
10 ' ---------- Programma KEPLERO  ----------
20 CLS
30 DIM P(2000,2)
40 '-------------------------------------------------
50 INPUT "Introduzione numero punti da calcolare";N
60 IF INT(N)<>N OR N<50 OR N>2000 THEN 50
70 INPUT " Introduzione valore massa centrale => ";M
80 IF M<=0 THEN 70
90 INPUT " Introduzione coordinate posizione => ";X,Y
100 INPUT " Introduzione valore velocita' (coordinate
    polari) => ";V,VTH
```



Questa ellisse allungata rappresenta l'orbita della cometa di Halley, calcolata con il programma KEPLERO. La cometa di Halley, il cui periodo di rivoluzione è di 76 anni, passerà al perielio nel 1986.

```
110 IF V(=0  THEN 100
120 INPUT " Introduzione valore intervallo elementare
    tempo di calcolo =)";DT
130 IF DT(=0  THEN 120
140 '-------------------------------------------------
150 CLS
160 CURSOR (30,15)
170 PRINT "ASPETTA, STO CALCOLANDO!"
180 '-------------------------------------------------
190 VX=V*COS (VTH*3.141592654# 180)
200 VY=V*SIN(3.1415926554#/180*VTH)
210 R= X^2+Y^2
220 TH=ATN(Y/X)
230 IF X(0 AND Y)0  THEN TH=TH+3 141592654#
240 IF X(0 AND Y(0  THEN TH=TH-3.141592654#
250 VX=VX-6.67E-11*M/R*COS(TH)*DT/2
260 VY=VY-6.67E-11*M/R*SIN(TH)*DT/2
270 P(1,2)=Y
280 P(1,1)=X
290 '-------------------------------------------------
300 FOR A=2 TO N
310 P(A,1)=P(A-1,1)+VX*DT
320 P(A,2)=P(A-1,2)+VY*DT
330 R=P(A 1)^2+P(A,2)^2
340 TH=ATN(P(A,2)/P(A,1))
350 IF P(A,1)(0 AND P(A,2)(0 THEN TH=TH+3 141592654#
360 IF P(A,1)(0 AND P(A,2)(0 THEN TH=TH-3 141592654#
370 VX=VX-6 67E-11*M/R*COS(TH)*DT
380 VY=VY-6.67E-11*M/R*SIN(TH)*DT
390 '-------------------------------------------------
400 IF (DT*5E+08/R^ 6)1.35 THEN 470
410 IF (DT*5E-08/R^ 6)1.35 THEN 500
420 '-------------------------------------------------
430 DT=DT*2
440 VX=VX-6.67E-11*M/R*COS(TH)*DT/4
450 VY=VY-6.67E-11*M/R*SIN(TH)*DT/4
460 GOTO 500
470 DT=DT/2
480 VX=VX+6.67E-11*M/R*COS(TH)*DT/2
490 VY=VY+6.67E-11*M/R*COS(TH)*DT/2
500 NEXT
510 '-------------------------------------------------
520 FOR A=1 TO N
530 IF MAXX(P(A,1) THEN MAXX=P(A,1)
540 IF MINX)P(A,1) THEN MINX=P(A,1)
550 IF MAXY(P(A 2) THEN MAXY=P(A,2)
560 IF MINY)P(A,2) THEN MINY=P(A,2)
570 NEXT
580 '-------------------------------------------------
590 CLS
600 AS=MAXX-MINX:OD=MAXY-MINY
610 IF AS=2*OD THEN 670
620 IF AS)2 *OD  THEN 640
630 AS1=(2*OD-AS)/2:MINX=MINX-AS1:MAXX=MAXX+AS1:GOTO 650
640 O1=(AS-2*OD)/4:MINY=MINY-O1:MAXY=MAXY+O1
650 AS=(MAXY-MINY)/2*1 32:O1=(MAXY+MINY)/2:MINY=O1
    -AS MAXY=O1+AS
660 '-------------------------------------------------
670 SCALE MINX,MAXX,MINY,MAXY
680 CIRCLE(0,0),(MAXX-MINX)/80:PAINT (0,0)
690 FOR A=1 TO N-1
700 LINE -(P(A,1),P(A,2))-(P(A+1,1),P(A+1,2))
710 NEXT
720 END
```



Anche questa orbita, come quella mostrata all'inizio dell'articolo, è stata calcolata con il programma KEPLERO: è un esempio di orbita iperbolica realizzata con un oggetto che passa vicino alla massa centrale con una velocità tale da farne un sistema libero, cioè non legato alla massa centrale stessa.

Analisi della struttura del programma KEPLERO

Le istruzioni dalla 50 alla 130 rappresentano l'introduzione dei dati.
Le istruzioni dalla 190 alla 260 effettuano uno "sfasamento" della velocità di metà intervallo di tempo (DT/2) in modo che il valore della velocità usato per il calcolo delle posizioni successive (con la formula $x = x_0 + V_x t$ e con la analoga per la y) corrisponda al valore medio dell'intervallo stesso. Ciò aumenta la precisione della simulazione.
Le istruzioni dalla 300 alla 380 effettuano il calcolo della posizione e della velocità negli intervalli successivi.
Segue quindi (dalla istruzione 400 alla 490) il controllo del valore DT dell'intervallo di tempo entro cui viene calcolata la posizione.
Questa variazione è necessaria soprattutto per orbite allungate poiché l'intervallo di tempo DT deve essere piccolo in prossimità della massa centrale (altrimenti la precisione andrebbe perduta) e grande nelle regioni lontane dalla massa centrale (altrimenti la memoria non sarebbe sufficiente e comunque i tempi di calcolo crescerebbero enormemente).
Il valore DT è stato scelto entro un intervallo avente per estremi i seguenti valori:

$$\frac{0{,}65}{5\cdot 10^8} \cdot R^{1,2}; \quad \frac{1{,}35}{5\cdot 20^8} \cdot R^{1,2}$$

il che ha dato buoni risultati.
Le singole posizioni vengono memorizzate su una matrice bidimensionale.
Segue poi la sezione relativa alla rappresentazione grafica dell'orbita: le istruzioni 520-570 determinano valori massimi e minimi delle coordinate relative alle varie posizioni, le istruzioni 600-650 determinano la scala in coordinate omogenee che contengano tutta l'orbita, e quelle 670-710 visualizzano la massa centrale e tracciano la traiettoria.

Conclusioni

Gli esempi visti, che a nostro giudizio sembrano sufficientemente ricchi e significativi, non sono che una spia delle grandi possibilità che sono offerte dall'informatica nell'ambito della didattica della fisica.
Sottolineiamo comunque che l'elaboratore non potrà mai sostituire completamente le esperienze eseguite direttamente nel laboratorio reale. L'esperienza diretta resta insostituibile, altrimenti, in sua mancanza, si correrebbe il rischio che lo studente si crei un mondo artificiale, staccato dalla realtà, il che è evidentemente contrario ai principi educativi più elementari. L'informatica si offre perciò come strumento valido e potente che integra gli strumenti tradizionali e che si dimostra indispensabile in tutti quei casi in cui l'esperimento reale si presenta lungo, difficile, costoso o addirittura semplicemente impossibile.

Visualizzazione di orbite newtoniane e campi elettrostatici

La prima figura mostra l'orbita terrestre, inizialmente equivalente a una circonferenza, calcolata con il programma KEPLERO. Se impostiamo i dati eseguendo alcune variazioni della velocità della Terra, a partire dai 30 km/s, quella effettiva, fino a 35, 38, 41 km/s, notiamo che corrispondentemente l'orbita si allunga sempre più.

Vediamo, più sotto, un altro esempio di applicazione per la fisica: si tratta di un programma che visualizza le linee di campo di un campo elettrostatico generato da due cariche puntiformi (programma FARADAY).

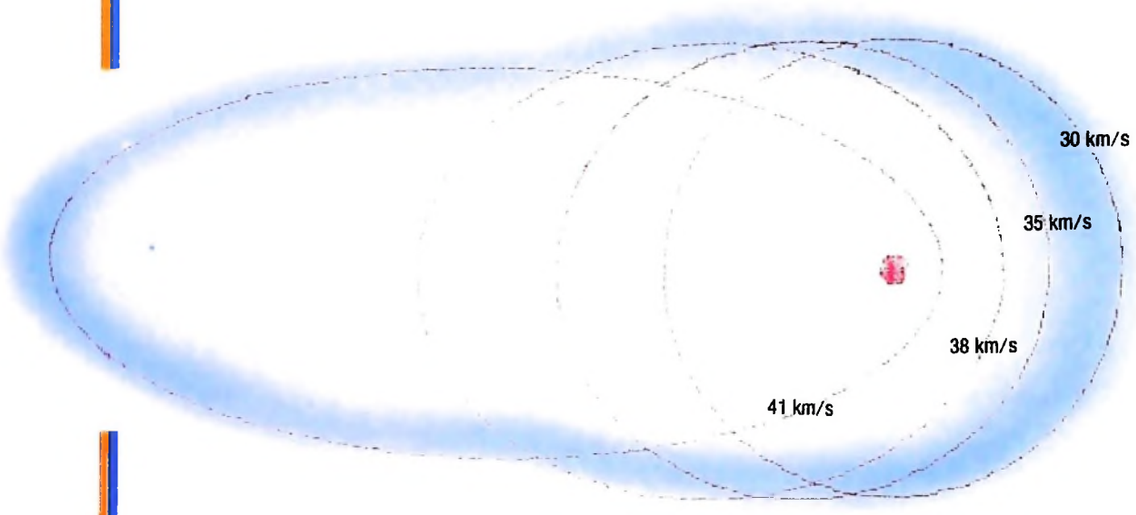
L'utente deve soltanto introdurre il valore di ciascuna carica (in Coulomb) e le coordinate delle rispettive posizioni.

L'elaboratore visualizza anzitutto le due cariche come piccoli cerchi in colore, quindi traccia 24 linee di campo a partire da ciascuna carica.

Le linee di campo partono da 24 punti che circondano ciascuna carica in modo regolare.

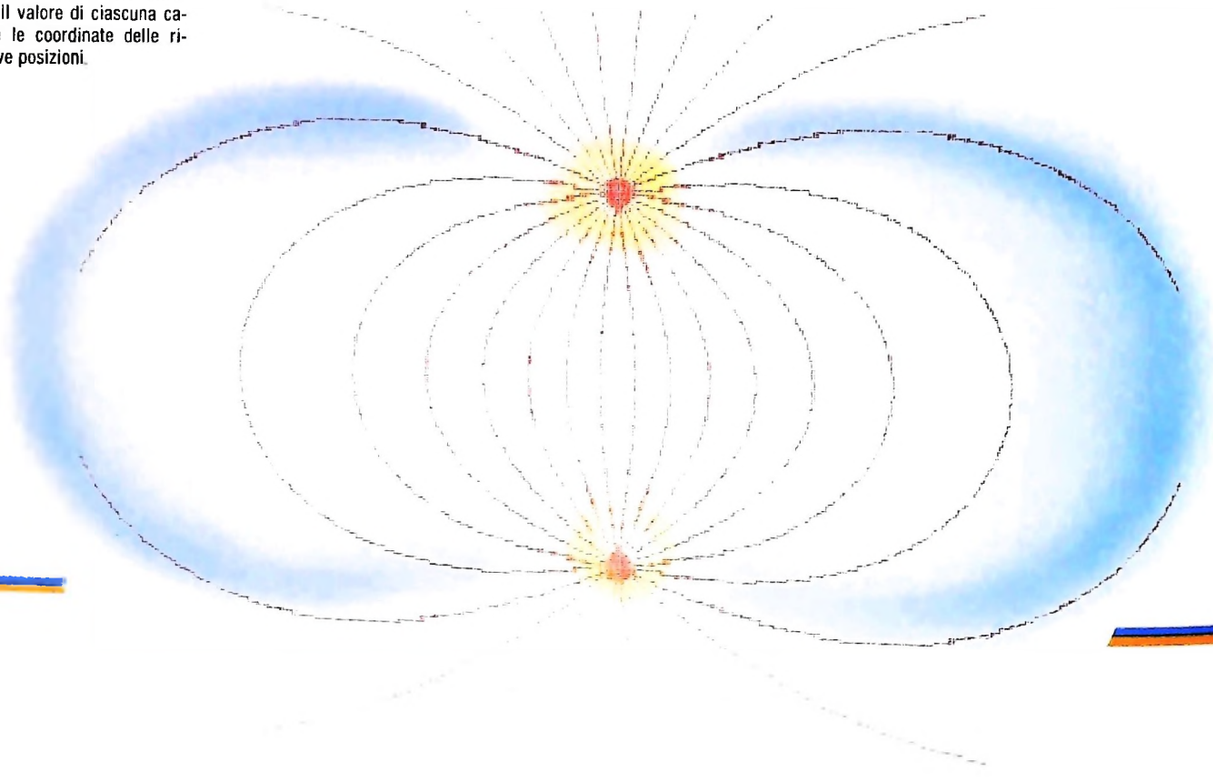
Il programma, come già detto, è stato scritto per l'elaboratore M20 tipo ST della Olivetti.

Ne presentiamo, in questa illustrazione, i risultati che ci sembrano più significativi e che riproducono esattamente quanto appare sullo schermo.



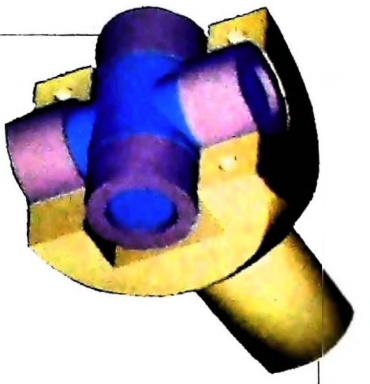
Una serie di orbite della Terra, calcolate con KEPLERO. Quella più a destra è l'orbita che si ottiene assegnando come velocità del pianeta quella reale; le altre si ottengono assegnando valori crescenti a tale velocità. Come si può vedere, la forma dell'orbita diventa sempre più ellittica.

Visualizzazione di un campo elettrostatico generato da due cariche puntiformi, mediante un programma scritto per l'elaboratore M20 ST della Olivetti. Per far girare il programma, l'utente deve solo introdurre il valore di ciascuna carica e le coordinate delle rispettive posizioni.

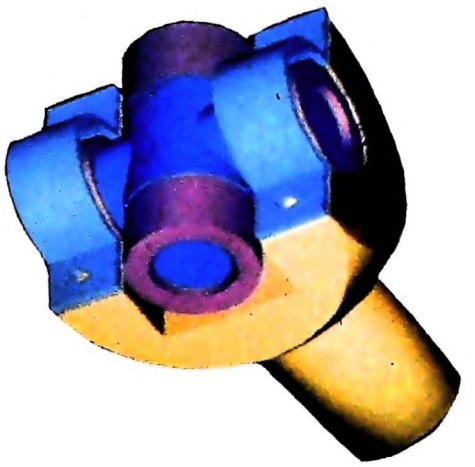


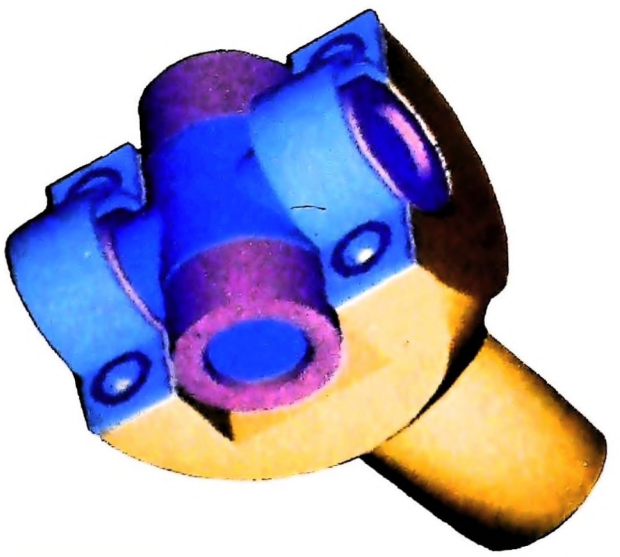
GRAFICA INTERATTIVA

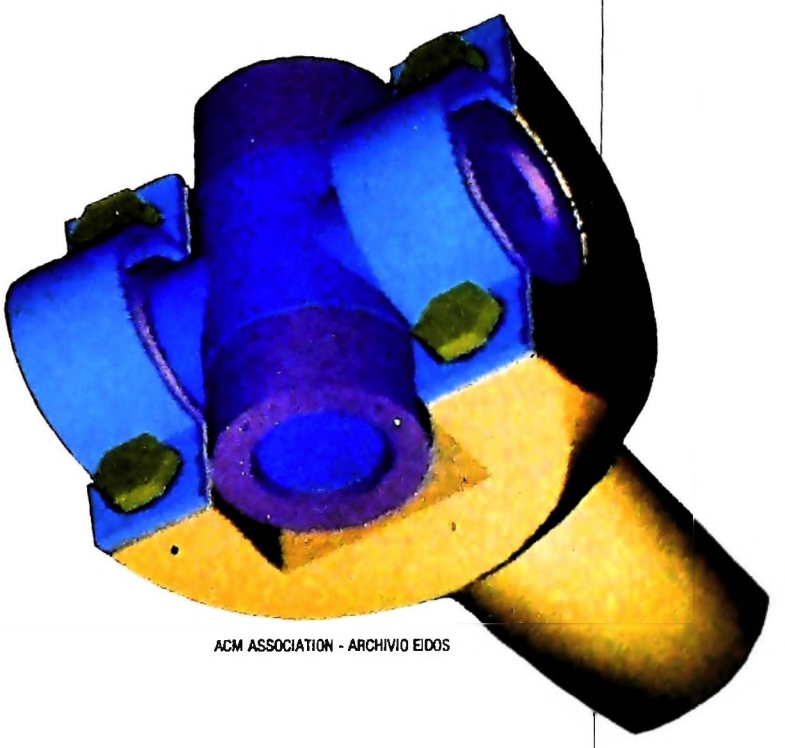
Un esempio di programma che permette di disegnare "direttamente" col computer.



In questa lezione vogliamo esporre un semplice esempio di elaborazione grafica interattiva. Innanzi tutto che cosa si intende con il termine interattivo? Semplicemente che esiste una possibilità di dialogo diretto fra utente e computer. Questa possibilità viene assicurata da una serie di istruzioni, ossia da un software, che può essere fornito in dotazione dal costruttore della macchina o che può essere creato dall'utente stesso. Lo scopo di tale software è quello di costituire una







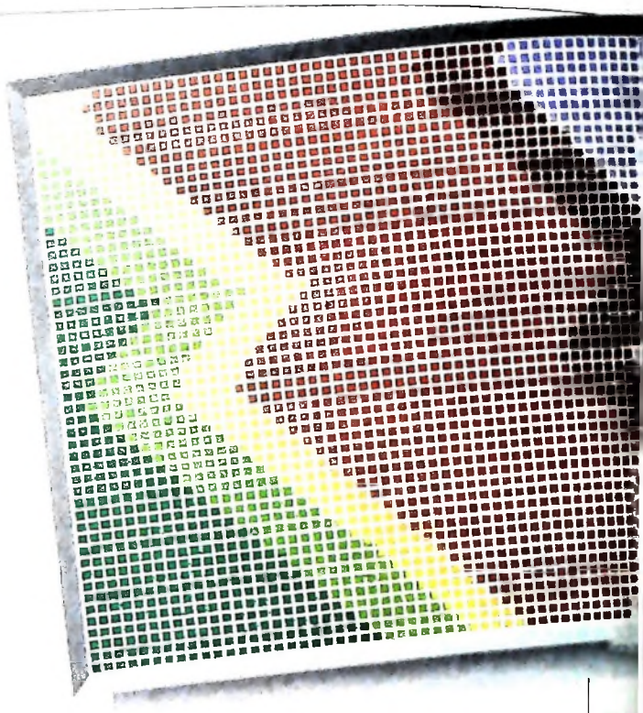
Un esempio di grafica interattiva. Nelle quattro immagini viene di volta in volta aggiunta una parte del pezzo meccanico e ne viene inoltre modificato il colore.

ACM ASSOCIATION - ARCHIVIO EIDOS

interfaccia fra utente e computer, proprio come un buon interprete simultaneo è in grado di mettere in comunicazione fra loro due persone che parlano lingue completamente diverse.

Nel nostro esempio, abbiamo creato un breve programma BASIC, che abbiamo chiamato PEN, il quale permette di utilizzare i tasti di posizionamento del cursore (le quattro freccette di M10) come una mano ideale, in grado di tracciare e cancellare delle linee sullo schermo. Prima di darne il listato vogliamo illustrare un concetto importante che viene utilizzato all'interno di PEN: ogni carattere ottenibile da tastiera è codificato all'interno di M10 utilizzando una particolare tabella di codici standard, detta ASCII; per esempio, il carattere B è codificato e riconosciuto dalla macchina con il codice ASCII "66". Questo ci permette di trattare qualsiasi carattere come un numero e quindi di effettuare dei test di controllo sul corrispondente codice. Per esempio, ci sarebbe impossibile far prendere una decisione a M10 secondo che il carattere battuto da tastiera sia la freccetta in alto o la freccetta in basso. Ciò è invece possibile se richiamiamo il codice ASCII della freccetta che, essendo un numero, è sottoponibile a un test decisionale. Nel programma PEN, di cui fornia-

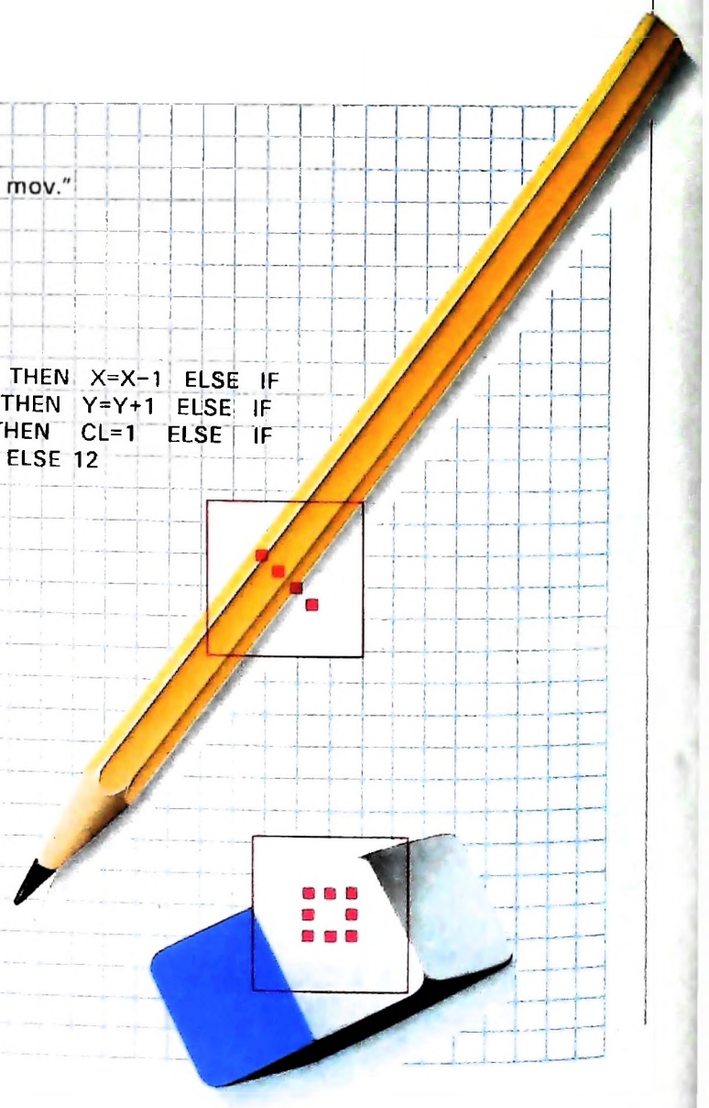
mo il listato, viene effettuata una serie di test di questo tipo. Nel commento verranno ulteriormente approfondite queste osservazioni.
Questo programma, scritto sfruttando le buone capacità grafiche di M10, permette di utilizzare, mediante semplici comandi, il display LCD (display a cristalli liquidi) come se fosse una lavagna.

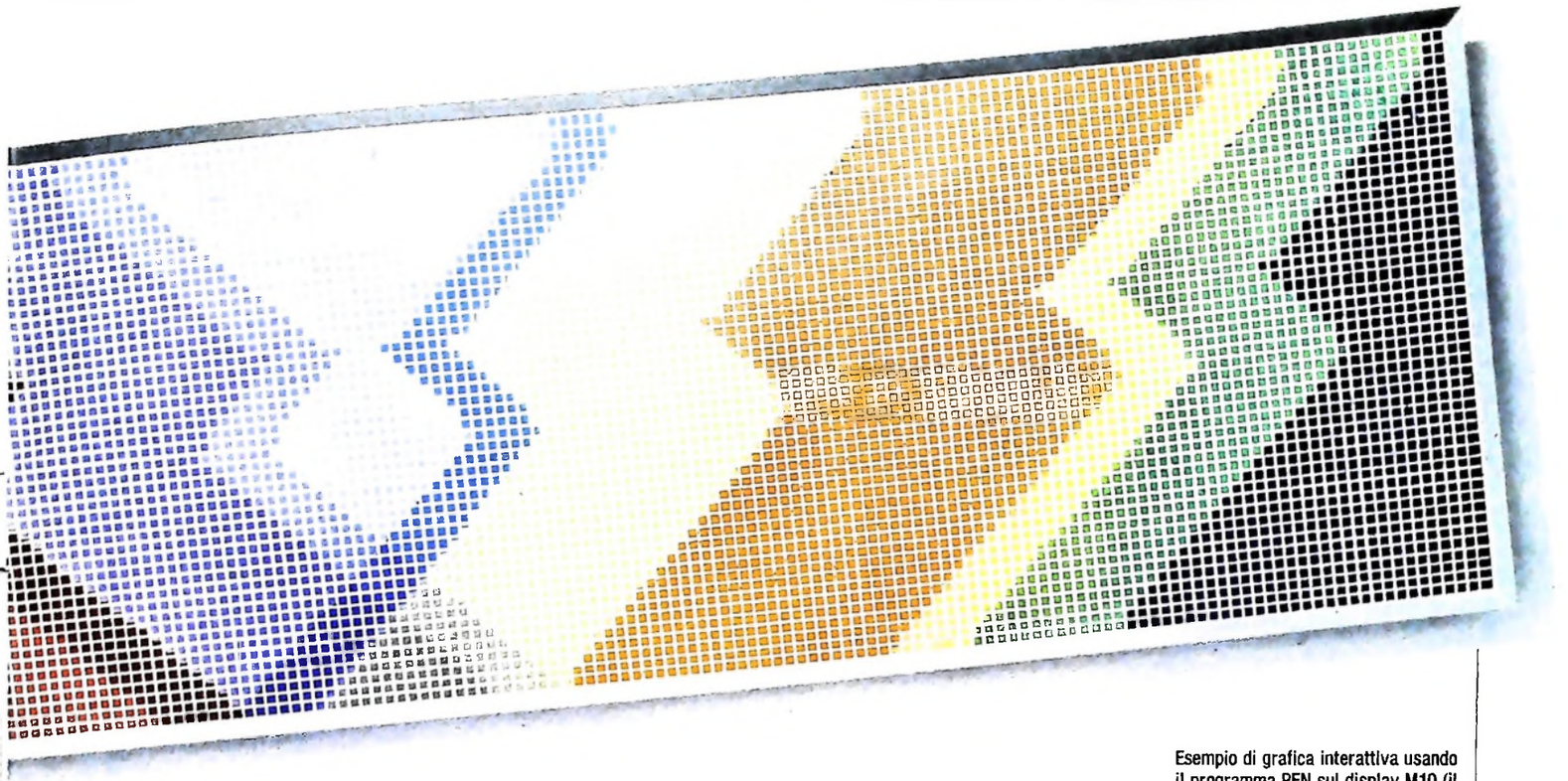


Commento al programma

Tutto il programma è basato sull'impiego di due particolari funzioni del BASIC:
ASC(A$): fornisce il codice ASCII di un carattere o del primo carattere di una generica stringa attribuita a una variabile (es. A$="B" ASC(A$)=66); nel programma PEN si sono utilizzati i seguenti codici ASCII:
28 = freccetta movimento cursore a destra
29 = freccetta movimento cursore a sinistra
30 = freccetta movimento cursore in alto
31 = freccetta movimento cursore in basso
103 = carattere "G"
109 = carattere "M"
115 = carattere "S"
99 = carattere "C"

```
5  REM**INIZIALIZZAZIONE E MENU**
10 CLS:PRINT"*PEN*G>omma,M>atita,C>ls S>top,        mov."
11 X=2 : Y=10 : CL=1
12 A$=""
13 A=X : B=Y
14 REM**LETTURE DA TASTIERAi**
15 A$=INKEY$ : IF A$=""THEN 15
20 REM
25  IF ASC(A$)=28 THEN X=X+1 ELSE IF  ASC(A$) =29  THEN X=X-1 ELSE IF
    ASC(A$)=30  THEN  Y=Y-1  ELSE IF ASC(A$)=31 THEN Y=Y+1 ELSE IF
    ASC(A$)=103 THEN CL=0 ELSE IF ASC(A$)=109  THEN  CL=1  ELSE  IF
    ASC(A$)=115 THEN END ELSE IF ASC(A$)=99 THEN 10 ELSE 12
26 REM
28 REM**CONTROLLO SULLE COORDINATE**
30 IF X<3 THEN X=3
31 IF X>236 THEN X=236
32 IF Y<10 THEN Y=10
33 IF Y>60 THEN Y=60
45 IF CL=0 THEN 60
46 REM
48 REM**DISEGNO E MOVIMENTO MATITA**
50 LINE(A-1,B-1)-(A+1,B+1),0,B F
52 LINE(X-2,Y-2)-(X+1,Y+1),1
54 PSET(X-1,Y-1,1)
55 GOTO 12
58 REM
60 REM**DISEGNO E MOVIMENTO GOMMA**
100 LINE(A-1,B-1)-(A+1,B+1),0,B F
105 LINE(A-2,B-2)-(A+1,B+1),0
110 LINE(X-1,Y-1)-(X+1,Y+1),1,B
130 GOTO 12
```





Esempio di grafica interattiva usando il programma PEN sul display M10 (il colore è stato aggiunto per differenziare le zone).

INKEY$: fornisce una stringa di un carattere che corrisponde al primo tasto battuto. Permette in pratica di far leggere un carattere in ingresso dalla tastiera con la semplice pressione del tasto (e cioè senza battere il tasto ENTER).
È molto usata nella forma seguente:
5 A$="" (A$ è la stringa vuota)
10 A$=INKEY$: IF A$="" THEN 10 (se nessun carattere è stato battuto si ritorna a chiederlo).
Sfruttando inoltre i codici ASCII delle quattro frecce in alto a destra, di solito usate per lo spostamento del cursore durante la stesura dei programmi (editing), è possibile controllare la posizione di un nuovo cursore generato dal programma sul display e conseguentemente, per mezzo delle istruzioni LINE e PSET, disegnare o cancellare.
Nel programma è da notare la "pesantezza" della linea 25, usata per aumentare la velocità decisionale in base alla lettura effettuata da tastiera.

Uso

Caricato il programma ed effettuato il consueto RUN (per esempio premendo F4) appare sullo schermo il titolo del programma e il menù d'uso le cui scelte vengono effettuate semplicemente digitando il primo carattere della parola corrispondente, senza il RETURN:

*PEN *G>omma,M>atita,C>ls,S>top, mov.

G>omma: permette di cancellare tutto ciò che appare sullo schermo all'interno dell'area del quadratino che la rappresenta.
M>atita: è utilizzata per il disegno vero e proprio ed è, come del resto la gomma, controllata nel suo spostamento dalle quattro frecce in alto a destra sulla tastiera di M10: la sua forma apparentemente strana deriva dalla necessità di minimizzare i tempi della visualizzazione durante il moto lungo lo schermo.
C>ls: la pressione di questo tasto permette la pulizia del display e il rientro nel programma.
S>top: premuto durante l'esecuzione del programma provoca il rientro in ambiente BASIC.
MOV.: questa dicitura, preceduta dallo spazio bianco, indica che battendo il tasto di spazio il cursore si può spostare senza modificare il disegno già realizzato.
Unico problema è il fatto che ripassando sopra a un disegno con il cursore in modalità M>atita se ne ottiene la cancellazione. Ciò è dovuto al particolare metodo di visualizzazione dell'M10 (pixel acceso, pixel spento), unito alla necessità di disegnare il cursore nella posizione corrente cancellandolo dalla precedente per simularne il movimento.

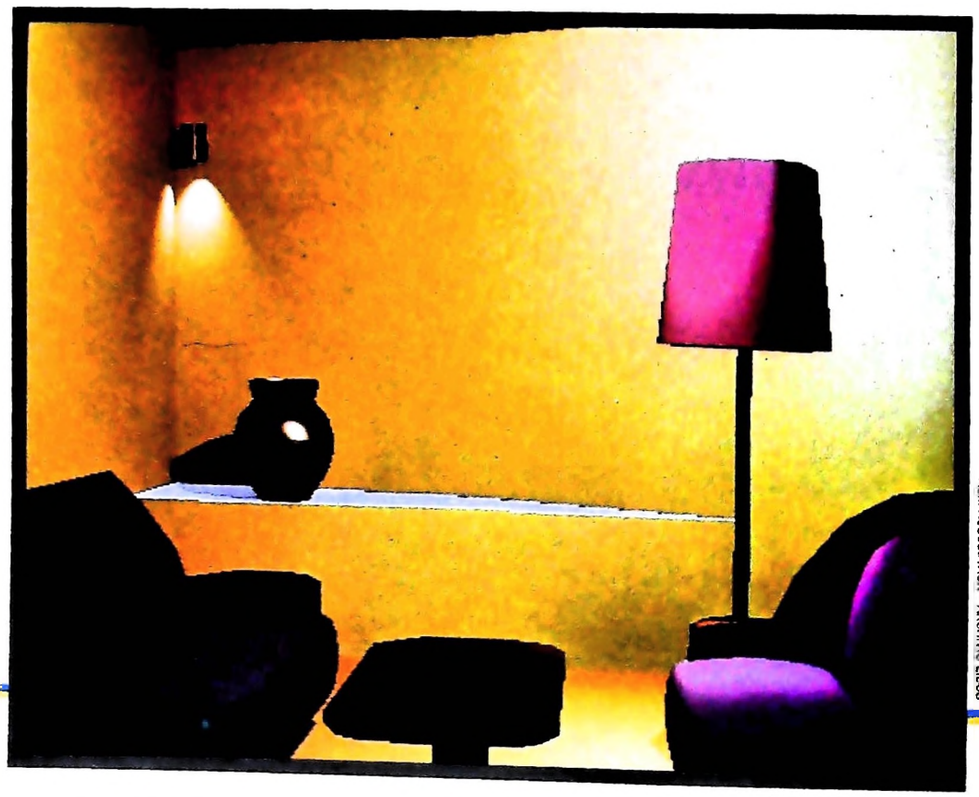
Ampliamenti

Fra i numerosi possibili ampliamenti del programma si consiglia di rendere disponibile il disegno di forme geometriche, utilizzando le opzioni BOX e FILL caratteristiche dell'istruzione LINE, prevedendo inoltre di poter spostare lungo lo schermo la forma così generata. Si prenda spunto dalle istruzioni del programma che governano il disegno ed il movimento della gomma (che è in pratica una forma geometrica in movimento lungo lo schermo), inserendo un'istruzione che permetta di dimensionare la figura mediante una specifica richiesta fatta all'utente del programma, come:
"Dimensione rettangolo?"
Altro interessante ampliamento è quello che permette di variare interattivamente la dimensione della forma geometrica

creata, semplicemente cambiando la posizione del cursore mediante le frecce di controllo. È importante per questo poter ricordare due posizioni del cursore, ad esempio mediante la pressione di un tasto supplementare di accettazione del comando, in modo tale da poter utilizzare l'istruzione LINE e l'opzione BOX (B) con le coordinate rilevate, tracciando così la forma nelle dimensioni desiderate.
Si consideri il seguente esempio pratico:
— viene letto il carattere che attiva la routine di inizio dell'operazione di ingrandimento;
— ci si ricorda ora la posizione del cursore al momento della battitura del tasto precedente;
— si sposta ora il cursore per rilevare il secondo punto;
— si batte un tasto per l'accettazione del comando;
— con una istruzione:
LINE(x1,y1)-(x2,y2),1,B
si ottiene il disegno della forma così creata.
La traduzione in BASIC dell'esempio indicato è inoltre un ottimo esercizio di analisi di un problema a partire dalle sue specifiche.

Architettura al calcolatore

La progettazione architettonica assistita dall'elaboratore è un settore in cui le applicazioni della grafica sono sempre più diffuse.
Qualcuno ha definito l'architettura come la disciplina dell'organizzazione dello spazio; cosa può esserci allora di meglio che creare un modello degli spazi architettonici e analizzarlo per verificarne gli aspetti più significativi?
In realtà l'uso del computer per la progettazione architettonica apre problematiche assai vaste che coinvolgono una revisione delle metodologie di progettazione per adattarle al nuovo mezzo. Alla base dell'uso del computer nella progettazione c'è la possibilità di sfruttare i mezzi di comunicazione che si fondano sulla grafica interattiva: deve cioè essere possibile simulare l'azione di un disegnatore al tavolo da disegno, mettendogli a disposizione matita, gomma, riga, compasso e tutto quanto serve per tradurre in segno un'idea.
Sarebbe però riduttivo pensare che l'aiuto che il computer può dare nella progettazione si esaurisca nel solo aspetto grafico; molto più interessante è poter integrare i differenti momenti di un ciclo progettuale utilizzando l'elaboratore non solo per aiutare nel disegno, ma anche per eseguire calcoli strutturali, simulazioni energetiche, verificare ipotesi tecnologicamente innovative, eseguire computi metrici.
Questa visione della progettazione come successione integrata o integrabile di momenti in cui intervengono specializzazioni differenziate ci permette di collocare al giusto posto gli elementi necessari. Per poter usare un elaboratore facendogli eseguire calcoli strutturali su progetti complessi ci rendiamo conto che nasce il problema di comunicare all'elaboratore la forma della struttura da calcolare. Analogo problema sorge nel caso della verifica ad esempio delle proprietà termiche dell'edificio, o per la costruzione del computo metrico. In tutti questi casi alla base delle varie operazioni si trova la forma, che deve poter essere analizzata dai diversi programmi che compendiano le specializzazioni normalmente distribuite tra ingegneri, architetti, geometri e altri specialisti. L'ossatura di un tale sistema integrato è data, ancora una volta, dalla capacità di rappresentazione grafica dell'informazione e dalla possibilità di comunicare in modo interattivo con l'elaboratore, costruendo passo passo il disegno di un progetto, prima in pianta, poi in sezione e infine, con l'alzato, realizzando un vero modello tridimensionale.
Costruire un progetto all'elaboratore, una volta che l'idea creativa è chiara nella mente dell'architetto, può essere visto come la simulazione di un vero e proprio processo costruttivo, in cui le componenti non sono veri mattoni o travi prefabbricate, ma la loro rappresentazione grafica, memorizzata nel computer insieme con le proprietà fisiche, geometriche e di costo.



ACM ASSOCIATION - ARCHIVIO EIDOS

Ricostruzione al computer di un interno. Sono presenti sorgenti di luce, ombre dirette e portate, riflessi.

—UN NUOVO MODO DI USARE LA BANCA.—

Conto corrente più

TANTI PENSIERI
IN MENO CON IL CONTO
CORRENTE "PIU'"
DEL BANCO DI ROMA.

Essere cliente del Banco di Roma vuol dire anche essere titolari del conto corrente "più". Un conto corrente più rapido: perché già nella maggior parte delle nostre filiali trovate gli operatori di sportello che vi evitano le doppie file.

Più comodo, perché potete delegare a noi tutti i vostri pagamenti ricorrenti: dai mutui all'affitto, dalle utenze alle imposte.

Più pratico, perché consente l'utilizzo del sistema di prelievo automatico Bancomat e l'ottenimento della carta di credito.

Più esclusivo, perché potete usufruire del servizio Voxintesi, attraverso il quale chiedere direttamente al nostro elaboratore il saldo del vostro conto corrente con una semplice telefonata: in qualsiasi ora come in qualsiasi giorno, anche festivo.

Più sicuro, perché con una minima spesa potrete assicurarvi contro furti e scippi mentre vi recate in banca o ne uscite.

Veniteci a trovare, ci conosceremo meglio.

BANCO DI ROMA
CONOSCIAMOCI MEGLIO.

Olivetti M10 vuol dire disporre del proprio ufficio in una ventiquattrore. Perché M10 non solo produce, elabora, stampa e memorizza dati, testi e disegni, ma è anche capace di comunicare via telefono per spedire e ricevere informazioni. In grado di funzionare a batteria oppure collegato all'impianto elettrico, M10 mette ovunque a disposizione la sua potenza di memoria, il suo display orientabile a cristalli liquidi capace anche di elaborazioni grafiche, la sua tastiera professionale arricchita da 16 tasti funzione.

Ma M10 può utilizzare piccole periferiche portatili che ne ampliano ancora le capacità, come il microplotter per scrivere e disegnare a 4 colori, o il registratore a cassette per registrare dati e testi, o il lettore di codici a barre. E in ufficio può essere collegato con macchine per scrivere elettroniche, con computer, con stampanti. Qualunque professione sia la vostra, M10 è in grado, dovunque vi troviate, di offrirvi delle capacità di soluzione che sono davvero molto grandi. M10: il più piccolo di una grande famiglia di personal.

PERSONAL COMPUTER OLIVETTI M10

L'UFFICIO DA VIAGGIO

Anche in leasing con Olivetti Leasing

olivetti

Se desiderate ricevere a casa vostra contrassegno da Olivetti M10 Punto di Vendita o Rivenditore, compilate questo coupon e speditelo a Olivetti, Divisione Personal Computer, Via Meravigli 12, 20123 Milano.

NOME COGNOME

VIA N.

CAP CITTÀ

TELEFONO